AFRL-RY-WP-TP-2007-1223, V2

DIFFUSION MAPS AND GEOMETRIC HARMONICS FOR AUTOMATIC TARGET RECOGNITION (ATR)
Volume 2: Appendices

Steven W. Zucker and Ronald Coifman

Yale University

NOVEMBER 2007
Final Report

Approved for public release; distribution unlimited.

See additional restrictions described on inside pages

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//signature//

GREGORY ARNOLD, Ph.D.
Project Engineer
ATR & Fusion Algorithms Branch
Sensor ATR Technology Division

//signature//

DEVERT W. WICKER, Ph.D.
Acting Chief, ATR & Fusion Algorithms Branch
Sensor ATR Technology Division
Sensors Directorate

//signature//

STEVEN P. WEBBER. LtCol, USAF
Deputy Chief, Sensor ATR Technology Division
Sensors Directorate

This report is published in the interest of scientific and technical information exchange and its publication does not constitute the Government’s approval or disapproval of its ideas or findings.

*Disseminated copies will show “//signature//” stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE ***(DD-MM-YY)*** November 2007	**2. REPORT TYPE** Final	**3. DATES COVERED *(From - To)*** 23 November 2004 – 31 October 2007

4. TITLE AND SUBTITLE
DIFFUSION MAPS AND GEOMETRIC HARMONICS FOR AUTOMATIC TARGET RECOGNITION (ATR)
Volume 2: Appendices

5a. CONTRACT NUMBER
FA8650-05-1-1800

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
62204F

6. AUTHOR(S)
Steven W. Zucker and Ronald Coifman

5d. PROJECT NUMBER
6095

5e. TASK NUMBER
04

5f. WORK UNIT NUMBER
60950416

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Yale University
Computer Science Dept. Program in Applied Mathematics
51 Prospect St.
New Haven, CT 06520-8285

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Sensors Directorate
Wright-Patterson Air Force Base, OH 45433-7320
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/RYAT

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-RY-WP-TP-2007-1223, V2

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Report contains color. See also Volume 1 (AFRL-RY-WP-TM-2007-1223, V1).

14. ABSTRACT

Geometric harmonics provides a framework for taking data in high-dimensional measurement spaces and embedding them in low dimensional Euclidean space according to a similarity measure. Euclidean coordinates then characterize the "manifold" on (or near) which the data live. Our goal in this project is to develop this manifold as a mechanism for integrating data from different sensors to facilitate automatic recognition. During the tenure of this research grant, we were able to formulate and complete the first series of experiments on embedded fusion. The resulting experiment on integrating voice and audio streams was extremely successful. This definitely revealed the potential for this approach and set the stage for further experiments.

The problem formulation has been completed and confirmed with an experiment on the integration of audio and video streams. Researchers at AFRL, Wright-Patterson Air Force Base, have received a first version of the software, and are attempting to apply it to radar signal interpretation.

15. SUBJECT TERMS
geometric, algorithms, target recognition, ATR

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	SAR	212	Gregory Arnold, Ph.D. **19b. TELEPHONE NUMBER** *(Include Area Code)* (937) 255-4039

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Geometric diffusions for the analysis of data from sensor networks

Ronald R Coifman[1], Mauro Maggioni[1], Steven W Zucker[1] and Ioannis G Kevrekidis[2]

Harmonic analysis on manifolds and graphs has recently led to mathematical developments in the field of data analysis. The resulting new tools can be used to compress and analyze large and complex data sets, such as those derived from sensor networks or neuronal activity datasets, obtained in the laboratory or through computer modeling. The nature of the algorithms (based on diffusion maps and connectivity strengths on graphs) possesses a certain analogy with neural information processing, and has the potential to provide inspiration for modeling and understanding biological organization in perception and memory formation.

Addresses
[1]Program of Applied Mathematics, Department of Mathematics, Yale University, 10 Hillhouse Avenue, New Haven, CT 06520, USA
[2]Department of Chemical Engineering, A-217 Engineering Quadrangle, Princeton University, Princeton, NJ 08544, USA
Corresponding author: Coifman, Ronald R (coifman@math.yale.edu)

Current Opinion in Neurobiology 2005, **15**:XXX–XXX

This review comes from a themed issue on
New technologies
Edited by Gero Miesenböck and Richard GM Morris

0959-4388/$ – see front matter

Introduction

Data processing and analysis has always been a vital component of scientific research; increasingly so in our times [1••–4••,5], when highly resolved sensing in space and time gives rise to huge, high-dimensional datasets. The same holds when the data are the result of fine-grained computational modeling, rather than sensor output. In neuroscience, there are myriad sources of very high dimensional data. Perhaps the simplest example is a single spike train, or a sequence of 100 to 10,000 such trains [6]. The situation becomes much more interesting (and much more complicated) when one considers evaluating the information in electrode arrays in, for example, the retina [7], the hippocampus [8,9] or the motor cortex [10]. Apart from these foundational questions, 'untangling the distributed code' (e.g. [11,12]) is now a key question for developing man–machine interfaces [10,11], and is not unlike related questions for the analysis of EEG and MEG signals. The techniques described here should be relevant to many of these tasks, both for developing processing algorithms and for determining the level of structure and intrinsic information in the signals. The additional feature of extracting higher order concepts from data computationally resonates with the way such concepts are extracted from data physiologically. We comment on some such tentative 'cognitive processing' features of our data processing algorithms.

The mathematical theory underpinning these new data analysis algorithms is that of harmonic analysis on sets of data represented as points lying in n-dimensional Euclidean space, R^n , and on graphs constructed using this data. These graphs, connecting data points in a way to be described below, are in a way reminiscent of the interconnectivity graphs of sensor nodes (or neurons) in which the strength of the connections represents a high affinity between nodes. The main challenge involving the analysis of such complex structures lies in the ability to explain the transition from local 'affinities' of massive sensor outputs, or data, to some higher order concepts, regions of influence and connectivities on a macroscopic scale. The mathematical theory described here leads to various computational methodologies useful in data analysis and machine learning and, as such, provides a powerful tool for empirical modeling.

One goal of this review is to present these developments in data analysis; a second goal is to provide some insight into mathematical processing mechanisms. These might be useful to the scientist studying empirical data processing and biological information processing in the formulation of potential models of neuronal organization (or sensor fusion) at different levels of granularity. Our approach gives rise to Markov processes on graphs constructed using the data; and uses spectral theory and eigenfunctions of these Markov processes [1••,2••], leading to a natural geometric organization of complex data sets, providing a 'nonlinear' principal component analysis. We remark in passing that the top eigenfunction, corresponding to the highest eigenvalue, for the Web graph provides the 'importance ranking' used by 'Google'® for webpage ranking, whereas the subsequent eigenfunctions provide a more detailed mapping. More importantly, we show how these eigenfunctions, viewed as a mathematical and computational tool, can be replaced by 'aggregates of nodes', equipped with a notion of multiscale affinity which can, in principle, be implemented biologically through various linking systems. This provides a potential theoretical mechanism for simple emergent organization and learning that might have biological relevance. Although related ideas appear in a variety of contexts of data analyses, such as spectral graph theory [13], manifold learning and nonlinear dimensionality reduction [14–17], we augment them by showing that the diffusion distances are key intrinsic geometric quantities linking spectral theory of Markov processes to the corresponding geometry of the data, relating localization in spectrum to localization in data space [2]. Existing dimensionality reduction techniques typically focus either on global or on local features of the data; our methodology integrates features at all scales in a coherent multiscale structure.

Geometric diffusions for global structure definition of data

In applied mathematics we often view ensembles of data as graphs with a large number of vertices, with each vertex being a data point (e.g. a visual stimulus), and edges connecting very similar data points (in an application-specific sense). For example, two visual stimuli could be considered similar if they excite a visual receptor in a very similar way.

Discovering large-scale structures and extracting information from such graphs is, in general, a very challenging task. Often the data are high-dimensional, that is, represented by long strings of numbers (vectors); however, physical or other constraints force the set of points or their probability densities to be intrinsically lower-dimensional, so they can, in principle, be described by a small number of degrees of freedom [1••,2••,14–17,18••,19••]. Our goal is to organize and process the data so as to reveal the low-dimensional structure. We use diffusion semigroups to generate various multiscale inference (or affinity) geometries (ontologies).

We show that appropriately selected eigenfunctions of Markov matrices describing local transitions, or affinities in the system, lead to coarse-grained, macroscopic structures at different scales.

In particular, the leading eigenfunctions enable a low dimensional geometric embedding of the dataset into a lower-dimensional Euclidean space, so that the ordinary Euclidean distance in the embedding space measures intrinsic diffusion (inference, affinity or relevance) metrics of the data.

The Euclidean correlation in R^n , for large n is, in general, not a good measure of affinity, except possibly for very close-by data points. This is the reason for the introduction of the 'closeness' parameter ε in the formula below. The premise is that the Euclidean distance provides a meaningful measure of 'affinity' for data lying closer than a cutoff distance quantified by this ε; and is meaningless for data beyond this cutoff. One of the main contributions is to find an embedding space such that the Euclidean distance in this space is truly representative of the closeness ('affinity') among the data.

Mathematical background

Think of a point X_i in Euclidean space as representing a string of outputs from a neuron labeled by i (data vector, sensor output stream, and so on). A matrix of local affinities can be constructed as:

$$A_\varepsilon = [X_i \cdot X_j]_\varepsilon := \exp\{-(1 - X_i \cdot X_j)/\varepsilon\}$$
$$\|X_i\| = 1 \quad .$$

The strength of such a data-correlation based affinity decays rapidly with the distance of outputs (other data affinities are possible, including chemical). We renormalize this matrix to a Markov matrix A (or more precisely A_ε), with sums of the entries of each row equaling one. A measures local similarities, and corresponds to one step of a random walk on the data [1••,2••,20]; its powers A^t correspond to propagation of the local similarities by the Markov process after t steps (time) of the random walk. This random walk on the data gives rise to a geometric diffusion (analogous to the derivation of the diffusion equation from Brownian motion). For large t, all similarities are integrated along all paths, yielding information about global structures in the data. Remarkably, these can be efficiently computed: let $\varphi_l(i) = \varphi_l(X_i)$ be the l^{th} eigenvector of A evaluated at data point i, satisfying $A\varphi_l(i) = \lambda_l^2\varphi_l(i)$ (λ_l^2 are arranged in decreasing order). Then

$$A^t(X_i, X_j) = \sum \lambda_l^{2t}\varphi_l(X_i)\varphi_l(X_j)$$
$$= a_t(i, j) \equiv a_t(X_i, X_j).$$

We consider the map

$$X_i^{(t)} \to (\lambda_1{}^t\varphi_1(X_i), \lambda_2{}^t\varphi_2(X_i), \ldots, \lambda_{\mathrm{m}}{}^t\varphi_m(X_i)) \qquad = \hat{X}_i^{(t)}$$

called the ‘diffusion map’, embedding into R^m at time t. The square of the ‘diffusion distance’ at time t, measuring ‘divergence’ between nodes i and j, is:

$$d_t^2(i, j) = a_t(i,i) + a_t(j, j) - 2a_t(i, j) = \sum_1^m \lambda_l^{2t}(\varphi_l(i) - \varphi_l(j))^2 = \left\|\hat{X}_i^{(t)} - \hat{X}_j^{(t)}\right\|^2.$$

For large t this can be computed very accurately using only the corresponding first few eigenfunctions, because only a few of the terms λ_l^{2t} are above the level of precision of interest (Figure 1). This provides a diffusion map embedding of output data into a new low-dimensional Euclidean space, converting diffusion distance on the data points into Euclidean distance in the embedding space.

As a first simple example of data reorganization provided by the diffusion embedding, we consider a sampled geometric hourglass surface, idealizing a set of data points with two weakly connected clusters, see Figure 2. We embed the point cloud into three-dimensional Euclidean space so that the diffusion distance in the original space can be computed as the ordinary Euclidean length of the chord connecting them in embedding space. Because the diffusion is slower through the bottleneck, the two components are farther apart in the diffusion metric.

In Figure 3, we illustrate the organizational ability of the diffusion maps on a collection of images given in random order. The inputs are 2-D gray scale pictures of the object in ‘3D’ in various positions, each viewed as a $32 \times 32 = 1024$ dimensional vector. To calculate the embedding, one constructs the Markov matrix as above, and computes the first few eigenfunctions. The top two eigenfunctions reveal the orientation of ‘3D’, and organize the data accordingly, see Figure 3.

Next, we organize a heterogeneous material, consisting of two component materials (nodes, represented by circles and crosses), possessing different conductivities (Figure 4). Although the gross statistics of circles and crosses are identical on both lobes, the left lobe happens to have more highly conductive links, which reduces the diffusion distance between its constituent nodes. The left-to-right bottleneck increases the diffusion distance between the two lobes, because there are fewer paths connecting the left and right lobe. The actual long-time affinity structure is described in terms of the eigenfunctions (Figure 4): on the left all points are tightly linked, whereas on the right they maintain some distance. The map has accounted for the preponderance of connections through all paths of all lengths between the nodes.

The next example (Figure 5) represents an organization of the configuration space of lip images that arise from a single speaker. No structure is assumed. The local similarity between images, viewed as high-dimensional vectors, organizes them as above in the first three diffusion coordinates. Different locations in the diffusion plot correspond to different clusters of strongly related lip images.

Dynamic learning through diffusion geometry

We now use these ideas to describe various learning methodologies in which the diffusion mechanism is iteratively adjusted to improve accuracy.

First, we generalize the basic affinity matrix to enable purely empirical and dynamical modeling and learning.

Assume that a data point set (sensor output, individual neuron output strings, and so on) has been generated by a process, the local statistical characteristics of which vary from location to location. For each point x, we view the neighboring data points as generated by a local unknown diffusion process, the probability density of which is estimated by $p_x(y) = c_x \exp(-q_x(x-y))$, where q_x is a quadratic form obtained empirically (for example by local principal component analysis [21]) from the data in a small 'neighborhood' of x.

We use the matrix $\sum_y p_x(y)p_z(y) = a(x,z)$ to model the corresponding data-driven diffusion. The distance defined by this kernel is $d(x,z) = (\sum_y |p_x(y) - p_z(y)|^2)^{1/2}$, which can be viewed as the natural distance on the 'statistical tangent space' to the point cloud.

In a dynamical learning situation we can start with a data point x, use its Euclidean neighborhood to define $p_x(y)$ at x, then find the z s that can be reached from x to compute locally $a(x,z)$. We then propagate a density in a neighborhood of x via powers of A, stopping when the propagation by diffusion slows down.

When labels are available, separating (a subset of) the data in different classes, the information they provide can be incorporated in p_x, by locally warping the metric so that the diffusion starting in one class stays in that class without leaking to others. This could be obtained, for example, by using any kind of local discriminant analysis [21] to build a local metric, the 'fast' directions of which are parallel to the boundary between classes and the 'slow' directions of which are transverse to the class boundaries. We also suggest that an iterative, partially supervised procedure can lead to good results in many practical situations.

In Figure 6 we represent a diffusion from labeled samples, from three different types of tissue, seeking to identify all related samples in the image. Here, each pixel has an absorption spectrum, with 128 spectral dimensions. The middle image shows the failure of conventional 'nearest neighbor' classification, whereas the diffusion distance yields a better classification.

Multiscale analysis of diffusion and spectral analysis

Our goal is to replace the analytic construction of the eigenfunctions by direct combinatorial link organizations. We show that the emergent organization discovered above with the help of the eigenfunctions can be translated into a multiscale hierarchical geometry of data points. This point of view can be used as a guide for theoretical processing models in biological systems.

The first few eigenfunctions of the matrix A (or equivalently, of the Laplacian on a graph [13]) detect and organize global structures on the data-based graph [1,16]. It is often the case, in biological and other complex systems, that several organizational structures exist at different 'scales'. Sensor outputs can be grouped (compressed) into ensembles at different scales of complexity, to perform tasks at different levels of complexity or abstraction, and integrating the tasks performed at lower levels of complexity.

We sketch a technique for constructing these sets of structures at different scales on a set of outputs or data, starting from the finest granularity, and building up to more complex structures, all inter-related at each scale and across scales, culminating in the global structures detected and described by the analysis with eigenfunctions described above. In the case of clouds of data points, this translates into a multiscale analysis of the cloud of points; at each scale we have a set of aggregates of points, and relationships among these groups are determined by a power of the diffusion operator at that scale. We claim (see [2]) that the embedding provided by the eigenfunctions can also be achieved by a hierarchical regrouping of data, using affinity at different diffusion time scales as a grouping mechanism.

The construction alluded to above is most easily explained in terms of conventional semantic analysis of text documents, each document being a data point. Each document has coordinates that represent the frequency of occurrence of words in it. We correlate only documents with strong similarity of vocabulary. Given a document x, we can build a folder around it of documents with strong immediate affinity (i.e. nearest neighbors). This becomes a folder at 'scale 1'. To obtain a folder at 'scale 2' we consider all documents, y, that are nearest neighbors to a nearest neighbor of x (i.e. they are linked by a chain of length 2 to x), and measure affinity as the sum of strength of all these chains of length 2 linking y to x; we keep only those, y, with strong affinity to form a folder at scale 2. We repeat this process for all chains of length 4 and less. One can easily build a directory structure of folders at all dyadic scales, with folders at a fixed scale being disjoint. From our point of view, every sensor (every neuron) can be viewed as a document for which a string of sensor outputs are the coordinates (elementary semantic content), whereas the folders are groups of outputs combining similar or highly related outputs at different resolution (or abstraction) levels. In Figure 7 the elementary documents are various 6x6 patches of the image in the first panel. The folders at different levels of resolution correspond to higher level features of the image.

To relate this description to a mathematical formulation we start by observing, as above (Figure 1), that the numerical rank of $(\mathrm{A}_{\varepsilon})^{t/\varepsilon}$ decreases rapidly as t increases. In particular, if we consider the expansions $a_t(x,y)=\sum \lambda_i^{2t/\varepsilon}\varphi_i(x)\varphi_i(y)$, for $t=\varepsilon 2^j$, obtained by successive squaring, then for any fixed precision the summation can be restricted to smaller and smaller sets of indices.

Secondly, the columns $a_t(x,y)$ of the matrix $(\mathrm{A}_{\varepsilon})^{t/\varepsilon}$ represent the probability of transition in t steps from x to y.

We can also interpret the x column of the matrix A^{2^j}, $a_{2^j}(x,y)$, as a rank of affinity between sensor (neuron) output x and sensor (neuron) output y at scale j, and the collection of points y, such that $a_{2^j}(x,y)>\delta$ could represent all sensor (neuron) outputs y similar to x.

We present a very simple method for obtaining a hierarchical 'sensor folder' (or 'neuron group') organization, as described above for the text documents. A minimal collection of clusters organizing the whole set of points at different levels of granularity is obtained as follows: let $\{\mathrm{x}_k^{j+1}\}$ be a maximal subcollection of points in $\{x_k^j\}$(key-points at scale j), such that $1/2 \le d_{2^j}(x_k^{j+1}, x_i^{j+1})$, where $\{\mathrm{x}_k^o\}$ are the original points. Then any point is at distance at most $1/2$ at scale j from one of the selected 'key-points' at that scale, enabling us to create a document folder labeled by the key-point. It is easy to modify this construction to obtain a tree of non-overlapping folders.

This construction, when applied to text documents (equipped with semantic coordinates), builds an automatic folder structure with corresponding key documents characterizing the folders.

A detailed, refined construction of scaling functions (columns of A^t) and wavelets representing this multiscale organization of the graph is provided in Coifman and Maggioni [2], and connections with related algorithms in numerical analysis in Brandt [22]. This analysis of aggregation at different times (and corresponding scales), enables us to perform multiscale wavelet analysis on manifolds and graphs in a natural way. Applications include compression of functions on the dataset, denoising of such functions, and learning (in the sense of classification and regression) of functions on the dataset. Although the description of the analysis given above refers only to organization of existing data, we point out that the tools developed also enable the incorporation of new data points into the structure in a consistent way, and the extension of functions modeled on the data to new sensor outputs [1••,2••,4••].

The multiscale construction enables structure to emerge at different scales as a function of connectivity. In Figure 7 we show several small patches from a simple image. If all patches are considered, edge filters (at the finer scales) and blob filters (at the coarser scales) naturally arise. Note the clear curvature in their structure [23]. Restricting the number of patches would result in more V1-like 'receptive-fields' [24–27].

Stochasticity and coherence

Global geometric diffusions can be applied to data driven by a Langevin equation [19••] that is used to model many biological systems [28–30], for example, stochastic unsynchronized neuronal pulse trains. The macroscopic probability density behavior of such systems is governed by the Fokker–Planck operator [19••], the eigenfunctions of which can be empirically approximated as described above, leading to efficient descriptions of likely, long-time probability configurations and geometries [2••,9,19]. The connections between Bayesian learning and Fokker-Planck equations date back to Verrelst [31] and references therein.

Diffusion wavelets and global diffusion have both been applied successfully to learning processes in a variety of (stochastic) environments, where an agent (e.g. robot) learns optimal behavior for achieving certain tasks from past experiences [18••].

Conclusions

Diffusion geometries can reveal structure in data at different levels of organization. Because many sources of data in neuroscience are high-dimensional, understanding their primary, low-dimensional intrinsic structure can be insightful. It has been indicated that image patch structure can suggest receptive field properties, and that different properties emerge at different levels. The intrinsic dimensionality can also be useful for efficient data analysis. Many applications of these techniques in neuroscience remain to be tried, from spike train analysis to olfaction and the electroencephalogram (EEG). But perhaps more exciting is the possibility that emergent structure across levels will open a theoretical door into cognitive neuroscience and memory organization.

Matlab scripts for the computations involved in diffusion maps and multiscale analysis of diffusion are available online [32] or upon request from M Maggioni.

Acknowledgements

The authors would like to thank Y Keller, S Lafon, AB Lee and B Nadler for providing some of the figures, and P Jones for useful comments during the preparation of the manuscript. Research partially supported by Defence Advanced Research Projects Agency. M Maggioni is partially supported by National Science Foundation grant DMS-0512050.

References and recommended reading

Papers of particular interest, published within the annual period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

••1. Coifman RR, Lafon S: **Diffusion maps.** *Appl Comp Harm Anal* 2005, in press.
The authors present an introduction to diffusion maps and their applications.

••2. Coifman RR, Maggioni M: **Diffusion wavelets**. Tech Rep YALE/DCS/TR-1303. *Appl Comp Harm Anal* 2005, in press.

The authors provide an in-depth presentation of the multiscale construction of diffusion geometries for multiscale analysis on graphs and manifolds.

••3. Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner FJ, Zucker SW: **Geometric diffusions as a tool for harmonic analysis and structure definition of data. Part I: Diffusion maps**. *Proc Nat Acad Sci* 2005, **102**: 7426-7431.

The authors provide an introduction to geometric diffusions with applications to analysis of data sets and simulated physical systems.

••4. Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner FJ, Zucker SW: **Geometric diffusions as a tool for harmonic analysis and structure definition of data. Part II: Multiscale methods**. *Proc Natl Acad Sci USA* 2005, **102**:7432-7437.

The authors provide a short introduction to the construction of multiscale diffusion geometries, and techniques for out-of-sample extension of Laplacian eigenfunctions and diffusion wavelets.

5. Donoho D: **Data! Data! Data! Challenges and opportunities of the coming data deluge**. *Michelson Memorial Lecture Series* 2001, available online at: http://www-stat.stanford.edu/~donoho/Lectures/AMS2000/AMS2000.html

6. Rieke F, Warland D, de Ruyter van Steveninck R , Bialek W: *Spikes, exploring the neural code*. MIT Press, 1997.

7. Warland D, Reinagel P, Meister M: **Decoding visual information from a population of retinal ganglion cells**. *J Neurophys* 1997, **78**: 2336-2350.

8. Zhang K, Ginzburg I,McNaughton BL, Sejnowski TJ: **Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells**. *J Neurophysiol* 1998, **79**:1017-1044.

9. Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA: **A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells**. *J Neurosci* 1998, **18**:7411-7425.

10. Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA: **Real-time prediction of hand trajectory by ensembles of cortical neurons in primates**. *Nature* 2000, **408**:361–365.

11. Donoghue J, Nurmikko A, Friehs G, Black M: **Development of neural motor prostheses for humans.** *Advances in Clinical Neurophysiology* (*Supplements to Clinical Neurophysiology*, Vol. 57) Editors: Hallett M, Phillips LH, Schomer DL, Massey JM. 2004.

12. Georgopoulos AP, Kettner RE, Schwartz AB: **Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of**

the direction of movement by a neuronal population. *J Neurosci* 1988, **8**:2928-2937.

13. Chung F: *Spectral Graph Theory*, (92). American Mathematical Society: CMBS-AMS series in Mathematics; 1997.

14. Ham J, Lee DD, Mika S: **Schölkopf: a kernel view of the dimensionality reduction of manifolds**. In *Proceedings of the XXI Conference on Machine Learning, Banff, Canada, 2004*.

The authors provide an overview of several nonlinear dimensionality reduction and manifold learning techniques, under the umbrella of kernel methods.

15. Roweis ST, Saul LK: **Nonlinear dimensionality reduction by locally linear embedding**. *Science* 2000, **290**:2323-2326.

16. Belkin M, Niyogi P: **Laplacian eigenmaps for dimensionality reduction and data representation**. *Neural Comp* 2003, **15**: 1373-1396.

17. Tenenbaum JB, de Silva V, Langford JC: **A global geometric framework for nonlinear dimensionality reduction**. *Science* 2000, **290**:2319-232.

••18 Mahadevan S, Maggioni M: **Value function approximation with diffusion wavelets and Laplacian eigenfunctions**. *Proc NIPS* 2005, in press.

Introduction to the application of diffusion wavelets and Laplacian eigenfunctions to Markov decision processes and learning.

••19. Coifman RR, Lafon S, Kevrekidis Y, Nadler B: **Diffusion maps, spectral clustering and reaction coordinates of dynamical systems.** *Appl Comp Harm Anal* 2005, in press.

The authors supply an interpretation of diffusion geometries for data generated by simulations of certain classes of physical systems with several examples.

20. Szummer M, Jaakkola T: **Partially labeled classification with Markov random walks.** *Advances in Neuronal Information Processing Systems* 2001, **14**: 945-952.

21. Hastie T, Tibshirani R, Friedman JH: *The Elements of Statistical Learning*. Springer-Verlag; 2001.

22. Brandt A: **Algebraic multigrid theory: the symmetric case.** *Appl Math Comp* 1986, **19**: 23-56.

23. Dobbins A, Zucker SW, Cynader M: **Endstopped neurons in the visual cortex as a substrate for calculating curvature.** *Nature* 1987, **329**:438-441.

24. Bell AJ, Sejnowski TJ: **The independent components of natural images are edge filters**. *Vision Research* 1997, **57**: 3327-3338.

25. van Hateren J, van der Schaaf A: **Independent component filters of natural images compared with simple cells in primary visual cortex**. *Proc R Soc London B* 1997, **265**: 259-366.

26. Olhausen BA, Field DJ: **Sparse coding with an overcomplete basis set: a strategy employed by V1?** *Vision Research* 1997, **37**:3311-3325.

27. Caywood MS, Willmore, B, Tolhust DJ: **Independent components of color natural scenes resemble V1 neurons in their spatial and color tuning**. *J Neurophysiol* 2004, **91**:2859-2873.

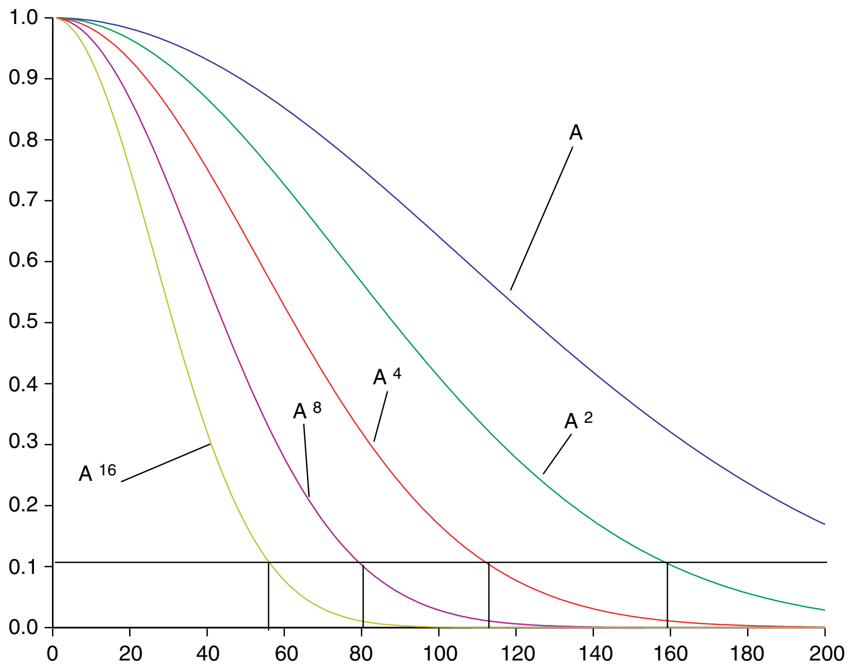
28. Rao CV, Wolf DM, Arkin AP: **Control, exploitation and tolerance of intracellular noise.** *Nature* 2002, **420**:231-237.

29.Vogels TP, Rajan K, Abbot LF: **Neural network dynamics**. *Annu Rev Neurosci* 2005, **28**:357-376.

30. Miesenböck G, Kevrekidis IG: **Optical imaging and control of genetically designated neurons in functioning circuits**. *Annu Rev Neurosci* 2005, **28**:533-563.

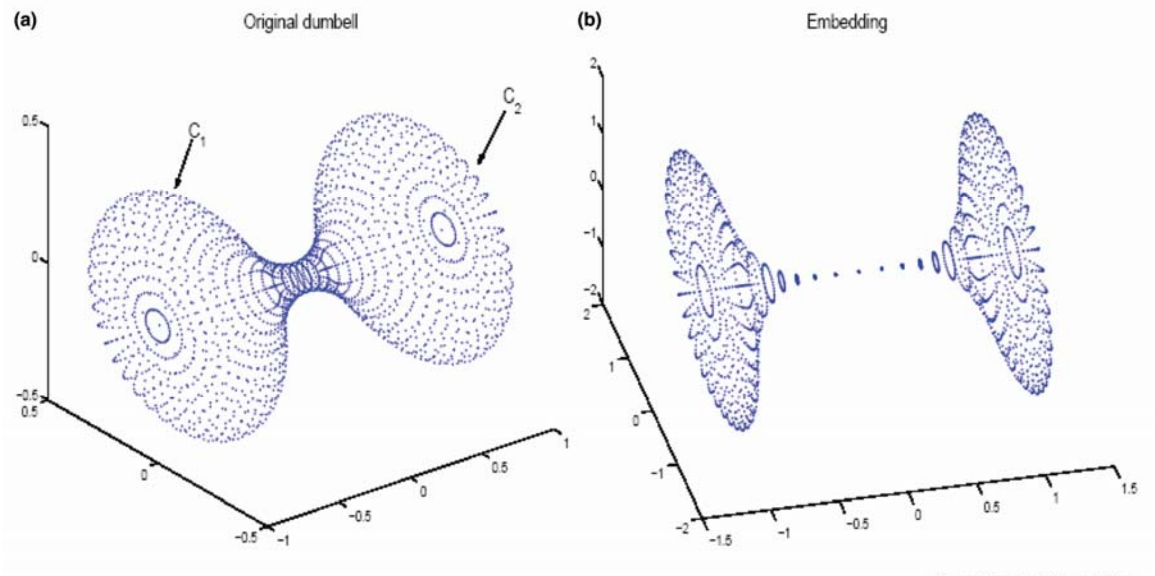
31. Verrelst H, Suykens J, Vandewalle J, De Moor B, **Bayesian learning and the Fokker-Planck machine.** In *Proceedings of the International Workshop on Advanced Black-box Techniques for Nonlinear Modeling, Leuven, Belgium, 1998*. 55-61.

32. Maggioni M: Homepage, URL: www.math.yale.edu/~mmm82



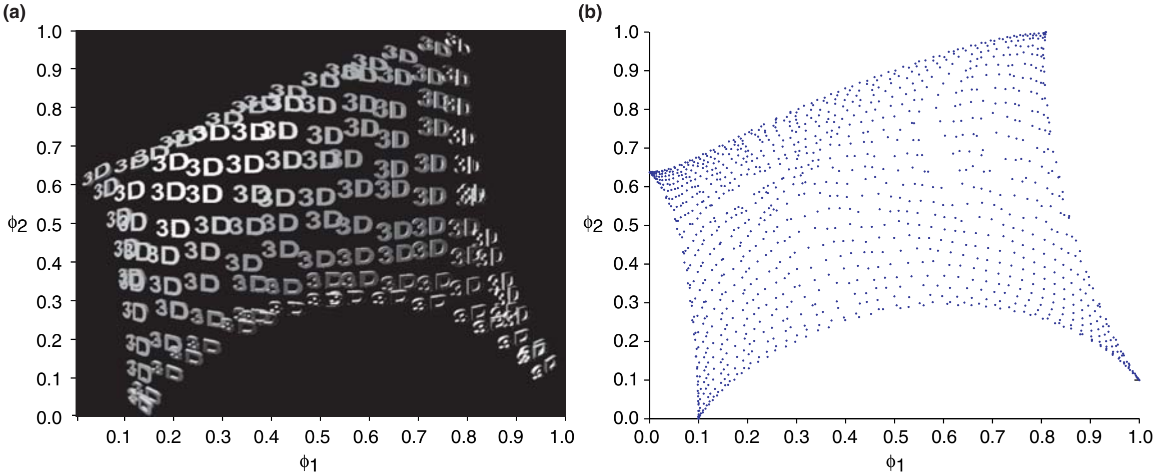
Current Opinion in Neurobiology

Figure 1
The spectra of powers of A. Some examples of the spectra of the dyadic powers of A. The x axis is the index of the eigenvalue, and the y axis the eigenvalue itself. Eigenvalues are positive and are arranged in nonincreasing order.



Current Opinion in Neurobiology

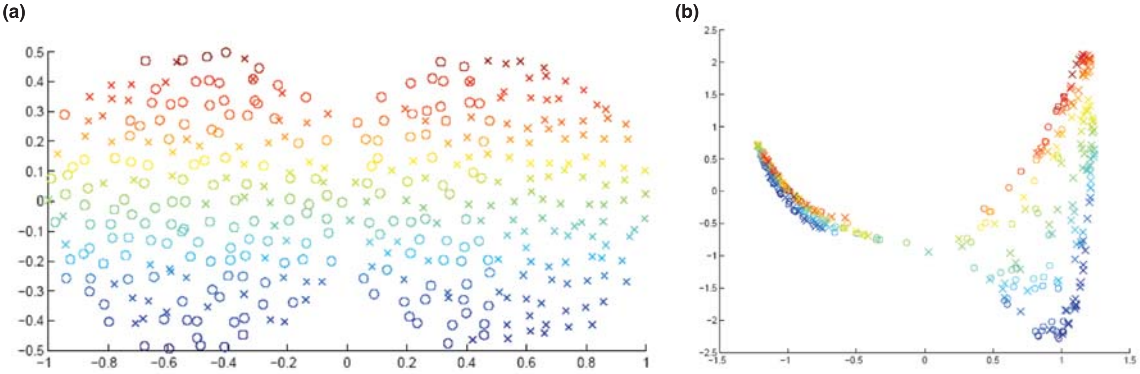
Figure 2
Diffusion embedding of a sampled hourglass manifold. **(a)** An original set of points sampled on a hourglass manifold, as a model for two weakly-connected clusters C1 and C2, and **(b)** their embedding using the eigenfunctions of the diffusion matrix A. The Euclidean distance in image in (b) is equivalent to large-time *t* diffusion distance on the original set of points in (a). The two 'clusters' get flattened and move further apart in the new space. The axes just provide a reference frame.



Current Opinion in Neurobiology

Figure 3

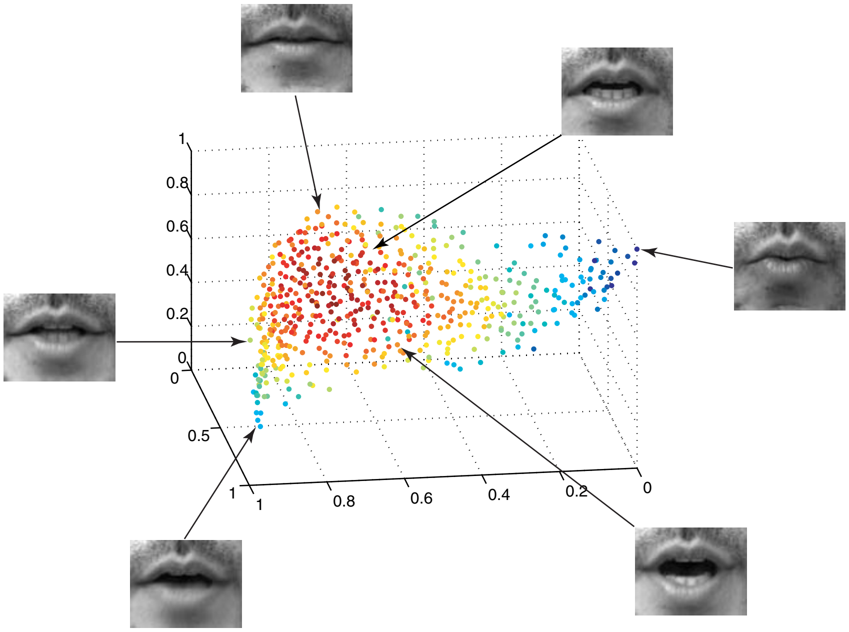
Diffusion embedding of a set of pictures of "3D". Organization emerging from a collection of images given in random order (data $=\{x_i\}$). **(a)** The images are displayed according to their location in the two-dimensional diffusion embedding $(\phi_1(x_i),\phi_2(x_i))$, displayed in **(b)**. The coordinates capture (perceive) the orientation of the picture in 3D.



Current Opinion in Neurobiology

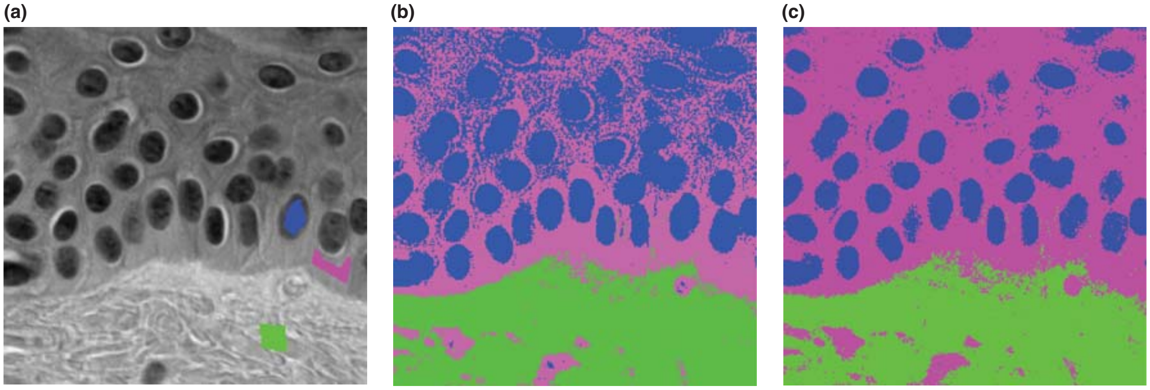
Figure 4

Diffusion embedding of a heterogeneous material. **(a)** A heterogeneous material and **(b)** its long-term diffusion embedding $(\phi_2(x_i),\phi_3(x_i))$. This structure could be interpreted as a map of trees (circles) and shrubs (crosses), with the links representing the probability of fire propagating among them. From (b) it is clear that the risk of fire propagating from top to bottom is higher on the left side of the forest. Color is included so that points can be matched across the two pictures.



Current Opinion in Neurobiology

Figure 5
Diffusion embedding of images of lips. The lip alphabet is learnt from a set of pictures of the lips of a speaker. The manifold structure and its parameters are parametrized by the three top eigenfunctions (axes in the figure) of the diffusion, and this parametrization can be used to lip-read. An interpretation of the low order eigenfunctions is openness of the mouth and exposure of teeth.



Current Opinion in Neurobiology

Figure 6
Classification of tissue types in a hyperspectral image through diffusion. **(a)** A slice of a hyperspectral image with three selected regions that correspond to three different biologically significant types of tissue: nuclei (blue), cytoplasm of epidermal cells (pink) and collagen in the underlying dermis (green). **(b)** Predictions of tissue type by a standard nearest neighbor classifier, trained on the set in (a). **(c)** Predictions made by the diffusion classifier described above, with the training set represented in (a).

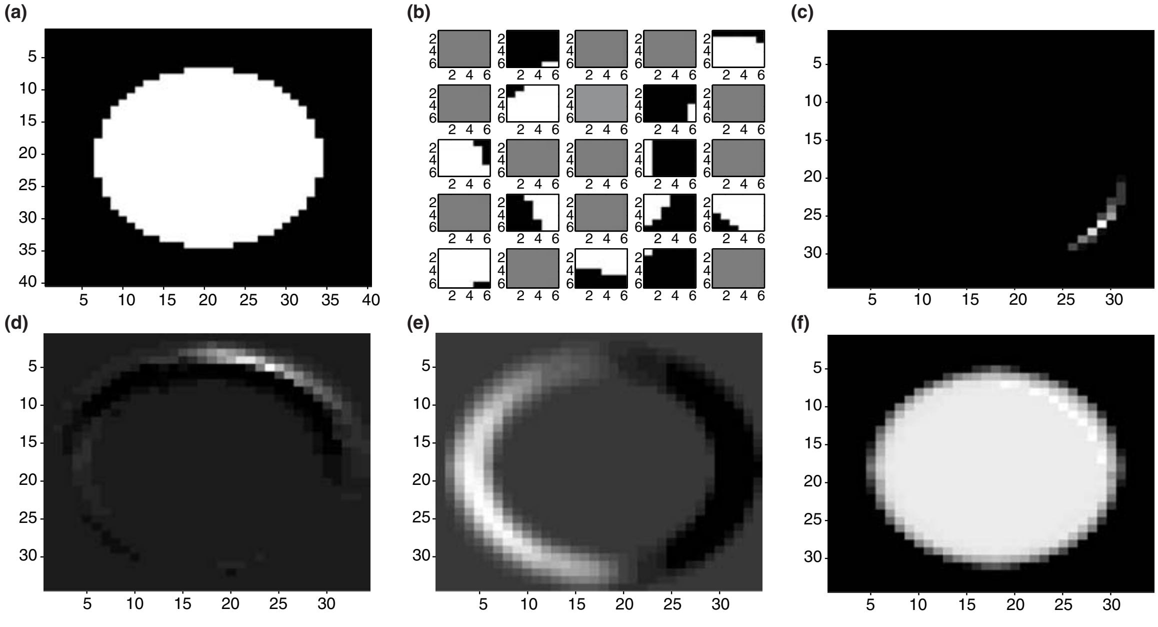


Figure 7
Multiscale folders. **(a)** Original picture. **(b)** A subset of 6×6 pixel patches extracted from the image. **(c)** A folder at scale 2 is a weighted aggregate of patches, representing a higher level feature. **(d)** Another folder at scale 2 is an edge detector. **(e and f)** Two folders at scale 3 that represent weighted aggregates of patches ('attributes' or 'features') at an even coarser scale.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."
"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Data Fusion and Multi-Cue Data Matching by Diffusion Maps

Stéphane Lafon[1], Yosi Keller[2] and Ronald R. Coifman[2]

Abstract

Data fusion and multi-cue data matching are fundamental tasks of high-dimensional data analysis. In this paper, we apply the recently introduced diffusion framework to address these tasks. Our contribution is three-fold. First, we present the Laplace-Beltrami approach for computing density invariant embeddings which are essential for integrating different sources of data. Second, we describe a refinement of the Nyström extension algorithm called “geometric harmonics”. We also explain how to use this tool for data assimilation. Finally, we introduce a multi-cue data matching scheme based on nonlinear spectral graphs alignment. The effectiveness of the presented schemes is validated by applying it to the problems of lip-reading and image sequence alignment.

Index Terms

Pattern matching, graph theory, graph algorithms, Markov processes, machine learning, data mining, image databases.

I. Introduction

The processing of massive high-dimensional data sets is a contemporary challenge. Suppose that a source s produces high-dimensional data $\{x_1, ..., x_n\}$ that we wish to analyze. For instance, each data point could be the frames of a movie produced by a digital camera, or the pixels of a hyperspectral image. When dealing with this type of data, the high-dimensionality is an obstacle for any efficient processing of the data. Indeed, many classical data processing algorithms have a computational complexity that grows exponentially with the dimension (this is the so-called “curse of dimensionality”). On the other hand, the source s may only enjoy a limited number of degrees of freedom. This means that most of the variables that describe each data points are highly correlated, at least locally, or equivalently, that the data set has a low intrinsic dimensionality. In this case, the high-dimensional representation of the data is an unfortunate (but often unavoidable) artifact of the choice of sensors or the acquisition device. Therefore it should be possible to obtain low-dimensional representations of the samples. Note that since the correlation between variables might only be local, classical global dimension reduction methods like Principal Component Analysis and Multidimensional Scaling do not provide, in general, an efficient dimension reduction.

First introduced in the context of manifold learning, eigenmaps techniques [1], [2], [3], [4] are becoming increasingly popular as they overcome this problem. Indeed, they allow one to perform a nonlinear reduction of the dimension by providing a parametrization of the data set that preserves neighborhoods. However, the new representation that one obtains is highly sensitive to the way the data points were originally sampled. More precisely, if the data are assumed to approximately lie on a manifold, then the eigenmap representation depends on the density of the points on this manifold [5]. This issue is of critical importance in applications as one often needs to *merge data* that were produced by the same source but acquired with different devices or sensors, at various sampling rates and possibly on different occasions. In that case, it is necessary to have a canonical representation of the data that retains the intrinsic constraints of the samples (e.g. manifold geometry) regardless of the particular distribution of the datasets sampled by different devices.

[1] Google Inc., stephane.lafon@gmail.com
[2] Department of Mathematics, Yale University, {yosi.keller, coifman-ronald}@yale.edu.

Another important issue is that of *data matching*. This question arises when one needs to establish a correspondence between two data sets resulting from the same fundamental source. For instance, consider the problem of matching pixels of a stereo image pair. One can form a graph for each image, where pixels constitute the nodes, and where edges are weighted according to the local features in the image. The problem now boils down to matching nodes between two graphs. Note that this situation is an instance of multi-sensor integration problem, in which one needs to find the correspondence between data captured by different sensors. In some applications, like fraud detection, synchronizing data sets is used for detecting discrepancies rather than similarities between data sets.

The out-of-sample extension problem is another aspect of the data fusion problem. The idea is to extend a function known on a training set to a new point using both the target function and the geometry of the training domain. The new point and the corresponding value of the function can then be assimilated to the training set. This is an essential component in any scheme that agglomerates knowledge over an initial data set and then applies the inferred structure to new data. Recently, Belkin *et al* have developed a solution to this problem via the concept of manifold regularization [6]. Earlier, several authors used the Nyström extension procedure in the Machine Learning context [7], [8] in order to extend eigenmap coordinates. In both cases, the question of the scale of the extension kernel remains unanswered. In other words, given an empirical function on a data set, to what distance to the training set can this function be extended ? In particular, given the spectral embedding of the data set, which kernel should be used to extend it?

By relating the frequency content of the target function on the training set to the extrinsic Fourier analysis, Coifman *et al* provide an answer to this question [9]. They developed the idea of "geometric harmonics" based on the Nyström extension at different scales, providing a multiscale extension scheme for empirical functions. We apply this concept to the extension of spectral embeddings and show that the extension has to be conducted using a specially designed kernel which differs from the eigenmap kernel.

In this article, we show that the questions discussed above can be efficiently addressed by the general diffusion framework introduced in [5], [10], [11]. The main idea is that, just like for eigenmaps methods, eigenvectors of Markov matrices can be used to embed any graph into a Euclidean space and achieve dimension reduction. Building on these ideas, the contribution of this paper is three-fold:

- First, we show that by carefully normalizing the Markov matrix, the embedding can be made invariant to the density of the sampled data points, thus solving the problem of data fusion encountered with other eigenmaps methods.
- Then, we address the problem of out-of-sample extension, and we explain how to adaptively extend empirical functions to new samples using the geometric harmonics. In particular this allows us to extend the diffusion coordinates to new data points.
- Last, we take advantage of the density-invariant representation of data sets provided by the diffusion coordinates to derive a simple data matching algorithm based on geometrical embeddings alignment.

The proposed scheme is experimentally verified by applying it to visual data analysis. First, we address the problem of automatic lip-reading by embedding the lips images using the Laplace-Beltrami eigenfunctions and deriving an automatic lip-reading scheme where new data is assimilated using geometric harmonics. Second, we demonstrate the multi-cue data matching aspect of our work by matching image sequences corresponding to similar head motions.

This paper is organized as follows: we start by recalling the diffusion framework, and the notion of diffusion maps in Section II-A. We then explain in Section II-B how to normalize the diffusion kernel in order to separate the geometry (constraints) of the data from the distribution of the points. We describe the out-of-sample extension procedure via the geometric harmonics in Section II-C and present a nonlinear algorithms for matching two data sets in Section II-D. Last, we illustrate these ideas by applying it to lip-reading and sequence alignment in Section III.

II. The diffusion framework

We start by reviewing the density-invariant embedding and out-of-sample extension schemes (previously introduced in [5] and [9]) in Sections II-B and II-C, respectively. To exemplify their applicability to high-dimensional data processing and learning, we apply them to derive a novel high-dimensional data alignment algorithm in Section II-D.

A. Diffusion maps and diffusion distances

Let $\Omega = \{x_1, ..., x_n\}$ be a set of n data points. In this section, we recall the diffusion framework as described in [5], [12], [13]. The main point of this set of techniques is to introduce a useful metric on data sets based on the connectivity of points within the graph of the data, and also to provide coordinates on the data set that reorganize the points according to this metric.

The first step in our construction is to view the data points $\Omega = \{x_1, ..., x_n\}$ as being the nodes of a symmetric graph in which any two nodes x_i and x_j are connected by an edge. The strength of this connection is measured by a non-negative weight $w(x_i, x_j)$ that reflects the similarity between x_i and x_j. The very notion of similarity between two data points is completely application-driven. In many situations however, each data point is a collection of continuous numerical measurements and, maybe after rescaling some of the features, it can be thought of as a point in a Euclidean feature space. In this case, similarity can be measured in terms of closeness in this space, and it is custom to weight the edge between x_i and x_j by $\exp(-\|x_i - x_j\|^2/\varepsilon)$, where $\varepsilon > 0$ is a scale parameter. This choice corresponds to the belief that the only relevant information lies in local distance measurements. Indeed, x_i and x_j will be numerically connected if they are sufficiently close. In diffusion kernels, graphs represent the structures of the input spaces, and the vertices are the objects to be classified. In addition, Belkin and Niyogi [2] explain that, in the case of a data set approximately lying on a submanifold, this choice corresponds to an approximation of the heat kernel on the submanifold. Last, in [5], it is shown that any weight of the form $h(\|x_i - x_j\|^2)$ (where h decays sufficiently fast at infinity) allows to approximate the heat kernel.

More generally, we allow ourselves to consider arbitrary weight functions $w(\cdot, \cdot)$ that verify the following two conditions[1], for all x and y in Ω:

- it is symmetric: $w(x, y) = w(x, y)$,
- it is pointwise non-negative: $w(x, y) \geq 0$.

This level of generality allows to take into account the case when data points are represented by a collection of categorical features. In this situation, it can be useful to employ a Gaussian kernel with a Hamming distance. But rather than to give a list of recipes, we would like to underline the fact that the choice of the weight function *should be entirely application-driven*. The weight function or kernel describes the first-order interaction between the data points as it defines the nearest neighbor structures in the graph. It should capture a notion of similarity as meaningful as possible with respect to the application, and therefore could very well take into account any type of prior knowledge on the data. The analysis of the data provided by the diffusion techniques depends heavily on the choice of the weight function. Last, note that the only real requirement for our technique to be applicable is to be able to define a *local* notion of similarity between the point. In other words, one must be able to answer the question of whether two points are (very) similar or not. This is a much simpler question than having to define a *global* distance between all pairs of points.

Following a classical construction in spectral graph theory [15], namely the normalized graph Laplacian, we now create a random walk on the data set Ω by forming the following kernel:

$$p_1(x, y) = \frac{w(x, y)}{d(x)} ,$$

where $d(x) = \sum_{z \in \Omega} w(x, z)$ is the degree of node x.

[1] Since $w(\cdot, \cdot)$ is supposed to represent the similarity between data points, it will be fair to assume that $w(x, x) > 0$

Since we have that $p_1(x, y) \geq 0$ and $\sum_{y \in \Omega} p_1(x, y) = 1$, the quantity $p_1(x, y)$ can be interpreted as the probability for a random walker to jump from x to y in a single time step. If P is the $n \times n$ matrix of transition of this Markov chain, then taking powers of this matrix amounts to running the chain forward in time. Let $p_t(\cdot, \cdot)$ be the kernel corresponding to the t^{th} power of the matrix P. In other words, $p_t(\cdot, \cdot)$ describes the probabilities of transition in t time steps.

The asymptotic behavior of this random walk has been used to find clusters in the data set [15], [16], [17], where the first non-constant eigenfunction is used as a classification function into two clusters. This was justified as a relaxation of a discrete problem of finding an optimal cut in a graph [16]. This approach was later generalized to using more eigenvectors in order to compute a larger number of clusters (see for instance [18], [19], [13]). Several papers form machine learning (in particular [14]) have underlined the connections and applications of the graph Laplacian to machine learning. Within the manifold learning community, the first few eigenvectors of this Markov chain have been employed for dimensionality reduction. In [20], [2] Belkin and Niyogi showed that when data is uniformly sampled from a low-dimensional manifold, the first few eigenvectors of P are discrete approximations of the eigenfunctions of the Laplace-Beltrami operator on the manifold, thus providing a mathematical justification for their use in this case.

If the graph is connected, then for $t = +\infty$ this Markov chain is governed by a unique stationary distribution ϕ_0 (see appendix I), which means that for all x and y,

$$\lim_{t \to +\infty} p_t(x, y) = \phi_0(y) \, .$$

The vector ϕ_0 is the top left eigenvector of P, *i.e.*, $\phi_0^T P = \phi_0^T$, and it can be verified that $\phi_0(y)$ is given by

$$\phi_0(y) = \frac{d(y)}{\sum_{z \in \Omega} d(z)} \, .$$

The pre-asymptotic regime is governed according to the following eigendecomposition [12]:

$$p_t(x, y) = \sum_{l \geq 0} \lambda_l^t \psi_l(x) \phi_l(y) \, , \tag{1}$$

where $\{\lambda_l\}$ is the sequence of eigenvalues of P (with $|\lambda_0| \geq |\lambda_1| \geq ...$) and $\{\phi_l\}$ and $\{\psi_l\}$ are the corresponding biorthogonal left and right eigenvectors (see appendix II for a proof). Furthermore, because of the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in the previous sum.

Unifying ideas from Markov chains and potential theory, the *diffusion distance* between two points x and z was introduced in [12], [5] as

$$D_t^2(x, z) = \sum_{y \in \Omega} \frac{(p_t(x, y) - p_t(z, y))^2}{\phi_0(y)} \, . \tag{2}$$

This quantity is simply a weighted L^2 distance between the conditional probabilities $p_t(x, \cdot)$, and $p_t(z, \cdot)$. These probabilities can be thought of as features attached to the points x and z, and they measure the influence or interaction of these two nodes with the rest of the graph.

By increasing t, one propagates the local or short-term influence of each node to its nearest neighbors, and this means that t also plays the role of a scale parameter. The comparison of these conditional probabilities introduces a notion of proximity that accounts for the connectivity of the points in the graph. In particular, unlike the shortest path, or geodesic distance, this metric is robust to noise as it involves an integration along all paths of length t starting from x or z. Empirical evidence supporting this claim is provided in [13]. The diffusion distance incorporates the notions of mixing time and clusterness used in classical graph theory [21].

The connection between the diffusion distance and the eigenvectors goes as follows (see appendix II):

$$D_t^2(x,z) = \sum_{l \geq 1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \,. \tag{3}$$

Note that ψ_0 does not appear in the sum because it is constant. This identity means that the right eigenvectors can be used to compute the diffusion distance. The diffusion distance therefore generalizes the use of the eigenvectors for finding bottlenecks and clusters in the graph [21], and extends this approach by taking into account more than just the second largest eigenvalue.

Furthermore, and as mentioned before, because of the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in the previous sum. Let $m(t)$ be the number of terms retained, and define the diffusion map

$$\Psi_t : x \longmapsto \left(\lambda_1^t \psi_1(x), \lambda_2^t \psi_2(x), \ldots, \lambda_{m(t)}^t \psi_{m(t)}(x) \right)^T \,. \tag{4}$$

This mapping provides coordinates on the data set Ω, and embeds the n data points into the Euclidean space $\mathbb{R}^{m(t)}$. In addition, the spectrum decay is the reason why dimension reduction can be achieved. This method constitutes a universal and data-driven way to represent a graph or any generic data set as a cloud of points in a Euclidean space. We also obtain a complete parametrization of the data that captures relevant modes of variability. Moreover, the dimension $m(t)$ of the new representation only depends on the properties of the random walk on the data, and not on the number of features of the original representation of the data. In particular, if we increase t, then $m(t)$ decreases and we capture larger-scale structures in the data.

B. Data merging using the Laplace-Beltrami normalization

We now direct our attention to the case when the original data points $\Omega = \{x_1, ..., x_n\}$ are assumed[2] to approximately lie on a submanifold $\mathcal{M}$ of $\mathbb{R}^d$. The so called “manifold model” holds for a large variety of situations, such as when the data is produced by a source controlled by a few free continuous parameters. For instance, consider the rotation of a human head and the lips motion of a speaker. We will study these examples later in this paper.

On the manifold $\mathcal{M}$, the data points were sampled with a density $q(\cdot)$ that may reflect some important aspect of the phenomenon that generated the data. For instance, as described in [12], for some data sets, the density is related to the free energy surface that governs the samples. On the other hand, the density may depend on the acquisition process and may be unrelated to intrinsic geometry or dynamics of the underlying phenomenon. In this situation, the distribution of the points is an artifact of the sampling process, and consequently, any “good” representation of the data should be invariant to the density.

Classical eigenmap methods provide an embedding that combines the information of both the density and geometry. For instance, with the Laplacian eigenmaps [2], one starts by forming the graph with Gaussian weights $w_\varepsilon(x,y) = \exp(-\|x-y\|^2/\varepsilon)$, and then constructs the random walk as described in the previous section. The eigenvectors are then used to embed the data set into a Euclidean space. It was shown in [5] that in the large sample limit $n \to +\infty$ and small scale $\varepsilon \to 0$, the eigenvectors tend to those of the Schrödinger operator $\Delta + E$, where Δ is the Laplace-Beltrami operator on $\mathcal{M}$, and E is a scalar potential that depends on the density q. As a consequence, the Laplacian eigenmaps representation of the data heavily depends on the density of the data points. In particular, it makes it impossible to fuse two data sets obtained from the same sensors but with different densities.

In order to solve this problem, we suggest to renormalize the Gaussian edge weights $w_\varepsilon(\cdot,\cdot)$ with an estimate of the density and to form the random walk on this new graph. This is summarized in Algorithm 1.

[2] Note that the density normalization that we describe in this section can be applied to more general structures such as a cloud of points. In this case, the diffusion coordinates will be invariant to the density of the points within this cloud.

Algorithm 1 Approximation of the Laplace-Beltrami diffusion

```
Start with a rotation-invariant kernel $w_\varepsilon(x,y) = h\left(\frac{\|x-y\|^2}{\varepsilon}\right)$.
Let
    $q_\varepsilon(x) \triangleq \sum_{y\in\Omega} w_\varepsilon(x,y)\,,$
    and form the new kernel
    $\widetilde{w}_\varepsilon(x,y) = \frac{w_\varepsilon(x,y)}{q_\varepsilon(x)q_\varepsilon(y)}\,.$    (5)
Apply the normalized graph Laplacian construction to this kernel, i.e., set
    $d_\varepsilon(x) = \sum_{z\in\Omega} \widetilde{w}_\varepsilon(x,y)\,,$
    and define the anisotropic transition kernel
    $p_\varepsilon(x,y) = \frac{\widetilde{w}_\varepsilon(x,y)}{d_\varepsilon(x)}\,.$
```

Let P_ε be the transition matrix with entries $p_\varepsilon(\cdot,\cdot)$. The asymptotics for P_ε are given in the following theorem.

Theorem 1: In the limit of large sample and small scales, we have

$$\lim_{\varepsilon\to 0}\lim_{n\to+\infty}\frac{I-P_\varepsilon}{\varepsilon} = \Delta\,.$$

In particular, the eigenvectors of P_ε tend to those of the Laplace-Beltrami operator on $\mathcal{M}$. We refer to [5] for a proof. A similar analysis for the case of a uniform density $q \equiv 1$ is provided in [2], [22].

This result shows that the diffusion embedding that one obtains from an appropriately renormalized Gaussian kernel does not depend on the density q of the data points of $\mathcal{M}$. This algorithm allows one to successfully capture the nonlinear constraints governing the data, independently from the distribution of the points. In other words, it separates the geometry of the manifold from the density.

C. Out-of-sample extension and the geometric harmonics

In most applications, it is essential to be able to extend the low-dimensional representation computed on a training set to new samples. Let Ω be a data set and Ψ_t be its diffusion embedding map. We now present the geometric harmonic scheme that allows us to extend Ψ_t to a new data set $\widetilde{\Omega}$. Since we need to relate the new samples to the training set, we will assume that Ω is a subset of a Euclidean space $\mathbb{R}^d$.

As mentioned in the introduction, the Nyström extension method is a popular technique employed in the machine learning community [7], [8] for the extension of empirical functions from the training set to new samples. As we discuss later, this method suffers from several drawbacks, and the scheme that we present in this section aims at solving these problems.

For the sake of completeness, we first recall the idea of Nyström extension [23]. We then point out its weaknesses, present our geometric harmonics extension scheme and explain how it solves the problems of the Nyström extension. Let $\sigma > 0$ be a scale of extension, and consider the eigenvectors and eigenvalues of a Gaussian kernel[3] of width σ on the training set Ω:

$$\mu_l\varphi_l(x) = \sum_{y\in\Omega} e^{-\|x-y\|^2/\sigma^2}\varphi_l(y) \text{ where } x \in \Omega\,.$$

[3] In order to simplify our presentation of the extension algorithm, we choose to work with a Gaussian kernel. In general, one can use any symmetric kernel with an exponential decay.

Since the kernel can be evaluated in the entire space, it is possible to take any $x \in \mathbb{R}^d$ in the right-hand side of this identity. This yields the following definition of the Nyström extension of φ_l from Ω to $\mathbb{R}^d$:

$$\overline{\varphi}_l(x) \triangleq \frac{1}{\mu_l} \sum_{y \in \Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \mathbb{R}^d . \tag{6}$$

Note that φ_l is being extended to a distance proportional to σ from the training set Ω. Beyond this distance, the extension numerically vanishes.

We now know how to extend the eigenfunctions of the kernel, and since these eigenfunctions form a basis of the set of functions on the training set, any function f on the training set can be decomposed as the sum

$$f(x) = \sum_l \langle \varphi_l, f \rangle \, \varphi_l(x) \text{ where } x \in \Omega \, ,$$

and we can define the Nytström extension of f to the rest of R^d to be

$$\overline{f}(x) \triangleq \sum_l \langle \varphi_l, f \rangle \, \overline{\varphi_l}(x) \text{ where } x \in \mathbb{R}^d . \tag{7}$$

This scheme seems very attractive, but it raises the question of the choice of the kernel of extension. In our exposition above, we considered a Gaussian of width σ, which implies that functions will be extended to a distance proportional to σ (the extension numerically vanishes beyond a multiple of this distance). Classically (see [7], [8]), when extending eigenmaps, the kernel being used for the extension is the same as the one employed for the computation of the eigenmaps on the training set. The focal point of the extension scheme that we now present is precisely to contradict this approach. Indeed, when computing the diffusion embedding or any other type of Laplacian eigenmap, one strives for using as small a scale $\sqrt{\varepsilon}$ as possible. The reason behind this is that, as shown in Theorem 1 and in [2], [22], [5], in the limit of small scales, the diffusion maps approximate the eigenvectors of the Laplace-Beltrami, allowing to capture the geometry of the underlying structure of the data set (such as the manifold geometry if there is an underlying manifold). On the contrary, when extending the diffusion coordinates off the training set, it is our interest to extend them as far as possible in order to maximize their generalization power. This has two consequences:

- The scale σ of the kernel used for extending should be as large as possible.
- This scale should not be the same for all functions that we are trying to extend. Indeed, we expect the scale of extension to be related to the complexity of the function to be extended. Low-complexity functions should be easy to extend very far from the training set. For instance the constant function on Ω is the simplest function on the training set, and should be extendable to the entire space R^d. On the contrary, a function with wild variations on Ω should have a limited range of extension, as their values off the training set are more difficult to predict.

These two observations give rise to the idea of adapting the scale of extension (and hence the kernel) to the function f to be extended. Therefore, all we need now is a criterion for determining the maximum scale of extension for f. To this end, fix $\sigma > 0$, and observe that in Equation 6, $\mu_l \to 0$ as $l \to +\infty$, which implies that the Nyström extension scheme described by Equation 7 is ill-conditioned. Of course, we can circumvent this problem if, in the same sum, we only retain the terms corresponding to μ_0/μ_l smaller than a given threshold $\eta > 0$:

$$\overline{f}(x) \triangleq \sum_{l:\mu_0 < \eta\mu_l} \langle \varphi_l, f \rangle \, \overline{\varphi_l}(x) \text{ where } x \in \mathbb{R}^d . \tag{8}$$

This way, the extension procedure has a condition number less than to η, and this variable plays the role of a regularization parameter. However, $\overline{f}$ and f no longer coincide on Ω, which means that $\overline{f}$ is no longer an extension of f. This is precisely the basis of decision about the scale σ: if it turns out that the difference between $\overline{f}$ and f on Ω is still acceptable (as measured by the reconstruction error), then this means that

f is extendable at a distance σ from Ω. Otherwise, it means that σ needs to be reduced. Indeed, if we decrease the value of σ, then the kernel of extension becomes finer, and its eigenvalues will decay more slowly. This allows the sum in Equation 8 to contain more terms, and $\overline{f}$ to be a better approximation of f on Ω. This geometric harmonics technique formalizes these observations into a scheme presented in Algorithm 2.

Algorithm 2 Multiscale extension scheme of diffusion coordinates via geometric harmonics

```
1: Let $\Omega \subset \mathbb{R}^d$ be the training set and $f = \psi_i : \Omega \rightarrow \mathbb{R}$ be the diffusion coordinate to be extended
   ($1 \leq i \leq m(t)$). Choose a condition number $\eta > 0$ and an admissible error $\tau > 0$.
2: Choose an initial (large) scale of extension $\sigma = \sigma_0$.
3: Compute the eigenfunctions of the Gaussian kernel with width $\sigma$ on the training set $\Omega$:
       $\mu_l \varphi_l(x) = \sum_{y \in \Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \Omega \,,$
   and expand $f$ on this orthonormal basis (on the training set $\Omega$):
       $f(x) = \sum_{l \geq 0} c_l \varphi_l(x) \text{ where } x \in \Omega \,.$
4: Compute the error of reconstruction on the training set that one obtains by retaining only the
   coefficients such that $\eta > \mu_0/\mu_l$ in the sum above:
       $Err = \left( \sum_{l:\, \eta \leq \mu_0/\mu_l} |c_l|^2 \right)^{\frac{1}{2}} \,.$
   If $Err > \tau$ then divide $\sigma$ by 2 and go back to point 3. Otherwise continue.
5: For each $l$ such that $\eta > \mu_0/\mu_l$, extend $\varphi_l$ via the Nyström procedure:
       $\overline{\varphi}_l(x) \triangleq \frac{1}{\mu_l} \sum_{y \in \Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \mathbb{R}^d \,,$
   and define the extension $\overline{f}$ of $f$ to be
       $\overline{f}(x) \triangleq \sum_{l \geq 0} c_l \overline{\varphi}_l(x) \text{ where } x \in \mathbb{R}^d \,.$
```

To summarize our ideas, if we increase the scale of extension, then the error of reconstruction on Ω will increase. Hence, the reconstruction error limits the maximal extension range. In fact, this limitation can be regarded as relating the complexity of the function on the training set to the distance to which it can be extended off this set. Here, the notion of complexity is measured in terms of frequency content on the training domain. For instance, a constant function has almost no complexity and one should be able to extend it in the entire space. If the number of oscillations of this function increases, then the distance to which one can extend it gets smaller. This illustrated on Figure 1. The geometric harmonics are therefore perfectly appropriate for extending the diffusion coordinates to new samples as higher-order and lower-order diffusion coordinates do not have the same number of oscillations.

D. Multi-cue alignment and data matching

The purpose of this section is to explain how the diffusion embedding can be efficiently used for data matching. Suppose that one has two data sets $\Omega_1 = \{x_1, ..., x_n\}$ and $\Omega_2 = \{y_1, ..., y_{n'}\}$ for which one would like to find a correspondence, or detect similar patterns and trends, or on the contrary, underline

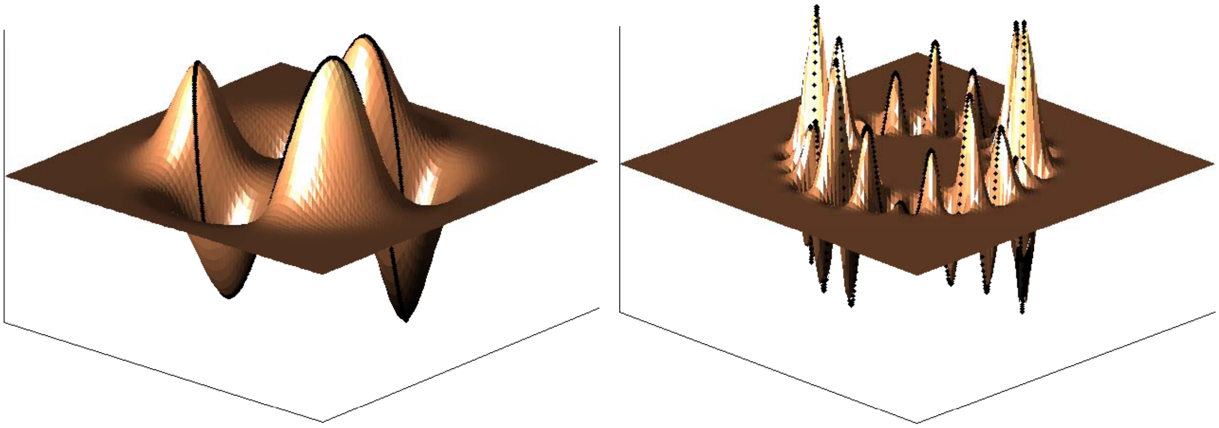


Fig. 1. Extension of two functions from the unit circle to $\mathbb{R}^2$. The function on the left is very smooth on the training set, and therefore can be extended far away from it. On the contrary, the function on the right oscillates much on the training set, and this limits its scale of extension.

their dissimilarity and detect anomalies. This type of task is very common in applications related to marketing, automatic machine translation, fraud detection or even counter-terrorism. However, working with the data in its original form can be quite difficult as the two sets typically consist of measurements of very different nature. For instance Ω_1 could be a collection of measurements related to wether in a given region, whereas Ω_2 could describe agriculture production in the same region. As a consequence, it is almost always impossible to directly compare the two data sets, simply because they might not be represented using the same type of features. The main idea that we introduce here is that the diffusion maps provide a canonical representation of data sets reflecting their intrinsic geometry. This new representation is based on the graph structure of a set, that is, the neighbor relationship between points, and not on their original feature representation. As a consequence, *instead of comparing the data sets in their original forms, it can be much more efficient to compare their embeddings*. In particular, if Ω_1 and Ω_2 are expected to have similar intrinsic geometry structures, then they should have similar embeddings.

There has been a body of work related to graph based manifold alignment. Gori et. al [24] align weighted and unweighted graphs by computing a ‘signature’ for each node that is based on repeated use of the invariant measure of different Markov chains defined on the data. The nodes/samples are then matched in two ways. First, in a one-by-one basis, where nodes with similar signatures are coupled. Second, in a globally optimal approach using a bipartite graph matching scheme. Ham et. al [25] align the manifolds, given a set of a-priori corresponding nodes or landmarks. A constrained formulation of the graph Laplacian based embeddings is derived by including the given alignment information. First, they add a term fixing the embedding coordinates of certain samples to predefined values. Both sets are then embedded separately, where certain samples in each set are mapped to the same embedding coordinates. Second, they describe a dual embedding scheme, where the constrained embeddings of both sets are computed simultaneously, and the embeddings of certain points in both datasets are constrained to be identical. The work of Bai et. al [26] presents a similar framework to our scheme. The ISOMAP algorithm is used to embed the nodes of the graphs corresponding to the aligned datasets, in a low-dimensional Euclidean space. The nodes are thus transformed into points in a metric space, and the graph-matching is recast as the alignment of point sets. A variant of the Scott and Longuet-Higgins algorithm is then used to find point correspondences. An approach to Many-to-Many alignment was presented in [27] by Keselman et. al. They aim to match corresponding clusters of nodes in both datasets, rather then match individual nodes. The datasets are embedded in a metric space using the Matousek embedding and sets of nodes are then aligned using the

Earth Mover's Distance, which is a distribution-based similarity measure for sets.

In the data alignment segment of our work, we resolve the alignment of datasets with a common low-dimensional manifold, but different densities, by incorporating the use of the density-invariant embedding. This issue was overlooked in previous works based on spectral embeddings [24], [25], [26], [27], although spectral and ISOMAPS embeddings are highly sensitive to the way the data points were originally sampled. Hence, the underlying assumption in [24], [25], [26], [27] that the low-dimensional embedding of datasets sharing a common low-dimensional manifold will be similar, might prove invalid.

In addition to dealing with the density issue, we present a semi-supervised algorithm for finding a one-to-one correspondence between two data sets. The scheme we introduce consists in aligning two graphs in a nonlinear fashion, based on a finite number of landmarks (matching points or nodes). The main idea is to lift each graph into the same diffusion space, and to align the resulting clouds of points using a simple affine matching[4]. The diffusion maps provide a nonlinear reduction of dimensionality, and therefore our scheme is appropriate for the alignment of high-dimensional data sets with low-intrinsic dimensionality. In addition, as explained in the previous sections, if we use the density-invariant diffusion maps, the alignment scheme will be insensitive to the different distributions of points of the two data sets.

As for the notations, suppose that we have $k < n, n'$ landmarks in each set, that is a sequence of k pairs $(x_{\sigma(1)}, y_{\tau(1)}), ..., (x_{\sigma(k)}, y_{\tau(k)})$ for which there is a known correspondence. This set of examples is the only prior information that we use in the algorithm. We assume that $x_{\sigma(1)} \neq x_{\sigma(2)} \neq ... \neq x_{\sigma(k)}$. The scheme given in Algorithm 3 computes a surjective function $g : \Omega_1 \rightarrow \Omega_2$ such that $g(x_{\sigma(1)}) = y_{\tau(1)}, ..., g(x_{\sigma(k)}) = y_{\tau(k)}$.

Algorithm 3 Nonlinear graph alignment

```
Start with $k$ landmarks $(x_{\sigma(1)}, y_{\tau(1)}), ..., (x_{\sigma(k)}, y_{\tau(k)})$.
Compute the diffusion embeddings $\{\widetilde{x}_1, ..., \widetilde{x}_n\}$ and $\{\widetilde{y}_1, ..., \widetilde{y}_{n'}\}$ of $\Omega_1$ and $\Omega_2$ where, for each set,
    the time parameter was chosen so that $k-1$ eigenvectors are retained. In other words, $\widetilde{x}_i$ and $\widetilde{y}_j$ both
    live in $\mathbb{R}^{k-1}$.
Compute the affine function $f : \mathbb{R}^{k-1} \rightarrow \mathbb{R}^{k-1}$ that satisfies the landmark constraints:
    $$f(\widetilde{x}_{\sigma(1)}) = \widetilde{y}_{\tau(1)}, ..., f(\widetilde{x}_{\sigma(k)}) = \widetilde{y}_{\tau(k)}\ .$$
Define the correspondence between $\Omega_1$ and $\Omega_2$ by
    $$g(x_i) = \arg\min_{y \in \Omega_2} \{\|f(x_i) - y\|\}\ ,$$
    where $x_i \in \Omega_1$,
```

The idea behind the scheme presented is to embed both data sets into the (same) diffusion space, and to use an affine alignment function f in the diffusion space. We assume that the choice of the kernels for computing the embeddings was already made by the user, and that they were selected in order to obtain meaningful graphs with respect to the application that the user has in mind. The number of eigenvectors used for the embedding is directly related to the number of landmarks, which in turns, represents the quantity of prior information for aligning. The larger the number of known constraints on the alignment, the larger the dimensionality of the aligning mapping. This is consistent with the fact that higher order eigenvectors capture finer structures. These observations pave the way for a general sampling theory for data sets. Indeed, the landmarks can be regarded as forming a subsampling of the original data sets. This subset determines the largest (or Nyquist) frequency used to represent the original set. This frequency is measured as the number of eigenvectors employed.

[4] We note that the alignment procedure can be automated for low-dimensional embeddings (up to R^3) by utilizing point matching schemes such as ICP [28] and Geometrical Hashing [29].

Note also that the affine function that we use for aligning induces a nonlinear mapping defined on lower dimensional embedding of the sets, and is even more nonlinear in the original space. It is possible to introduce more robustness to our scheme by embedding in a lower dimension than the number of landmarks, and to look for the best affine function that aligns the landmarks, where “best” is measured in a least-square sense.

III. Experimental results

A. Application to lip-reading

The validity of our approach is now demonstrated by applying it to lip-reading and sequence alignment, which are typical high-dimensional data analysis problems. From the statistical learning point of view, this example allows us to apply the ideas presented in the previous sections to three fundamental and related problems in the learning of high-dimensional data in general, and visual data in particular. First. we apply the diffusion framework to perform an efficient nonlinear dimensionality reduction. Second, we extend it to derive an intensity invariant embedding, essential for incorporating several data sources. Finally, we deal with the extension of a given embedding, computed on a given data set, to a new sample. This is the essence of a ‘learning’ schemes that associates knowledge obtained on a training set to a new set of samples.

Lip-reading has recently gained significant attention [30], [31], [32], [33], [34] and we now provide background and previous results in that field. The ultimate goal of lip-reading is to design human-like man-machine interfaces allowing automatic comprehension of speech, which in the absence of sound is denoted as lip-reading and the synthesis of realistic lip movement. The design of such a system involves three main challenges: first, the feature extraction, which aims at converting the images of the lips into a useful description, must be achieved with minimal preprocessing. Then, in order to be efficiently processed, the data must be transformed via a dimension reduction technique. Last, in order to assimilate new data for recognition, one must be able to perform data fusion.

Previous lip-reading schemes have mainly focused on the first two points. Concerning the feature extraction, some works [30], [34] analyze directly the intensity values of the input images, while others [35], [31] start by detecting curves and points of interest around the mouth whose locations are then used as features. The combination of audio-visual cues was used in [36] where the visual cues are the extracted lip contours which are tracked over time. We note that combining audio-visual is beyond the scope of this work and will be dealt by us in the future. Identifying, tracking and segmenting the lips is a difficult task and possible solutions include: active contours [37], probabilistic models [38] and the combination of multiple visual cues (shape, color and motion) [39] to name a few. In practice, one strives to use a simple preprocessing scheme as possible and in our scheme we employ a simple stabilization scheme discussed below.

Regarding the dimensionality reduction, several schemes have been used. Preliminary work employed linear algorithms such as the PCA and SVD subspace projections [35], [34]. For instance, Li *et al* [34] use a linear PCA scheme similar to the eigenfaces approach to face detection. Recognition is performed by correlating an input sequence with the eigenfeatures obtained from PCA. More recent schemes [30] utilize non-linear approaches such as the MDS [40]. Some of the techniques provide a general embedding framework for lipreading analysis [30], while others [34], [31] concentrate on a particular task such as phoneme or word identification. The work in [41] is of particular interest, since it is one of the first to explicitly formulate the lipreading problem as a “Manifold Learning” issue and tries to derive the inherent constraints embedded in the space of lip configurations. A Hidden Markov Model (HMM) is used to model a small number of words (names of four drinks) which define the Markov states and the manifold. The HMM is then used to recognize the drinks’ names where the input is given by tracking the outer lips contour using Active Contours. Utilizing both audio and visual information significantly decreased the error rate, especially in noisy environments. Kimmel and Aharon [30] applied the MDS scheme to visual lips representation, analysis and synthesis. A set of lips images is aligned and embedded in a two dimensional domain which is then sampled uniformly in the embedding domain to achieve uniform density. The pronunciation of each word is defined as a path over the embedding domain and used for visual speech recognition, by path matching. Lips motion synthesis is derived by computing the geodesic path over the embedding domain, where the start and end point are given as input. Anchor points in the low-dimensional embedding domain were then used to match the lips configurations of two different speakers.

Analysis of lip data constitutes an application where it is important to separate the set of nonlinear constraints on the data from the distribution of the points. As an illustration of the Laplace-Beltrami normalization as well as the out-of-sample extension scheme, we now describe an elementary experiment that paves the way to building automatic lip-reading machines, and more generally, machine learning systems.

We first recorded a movie of the lips of a subject reading a text in English. The subject was then asked to repeat each digit “zero”, “one”, ... , “nine” 40 times. A minimal preprocessing was applied to the recorded sequence. More precisely, it was first converted from colors to gray level (values between 0 and 1). Moreover, using a marker put at the tip of the nose of the speaker during the recording, we were able to automatically crop each frame into a rectangular area around the lips. Each of these new frames was then regarded as a point in $\mathbb{R}^{140\times 110}$, where 140×110 is the size of the cropped area.

The first data set, consisting of approximately 5000 frames, corresponds to the speaker reading the text. This set was used to learn the structures of the lip motion. More precisely, we formed a graph with Gaussian weights $\exp(-\|x_i - x_j\|^2/\varepsilon)$ on the edges between all pairs of points, where the distance $\|x_i - x_j\|$ was merely calculated as the Euclidean L^2 distance between frames i and j. The scale $\varepsilon > 0$ was chosen by looking at the distribution of the distances from each point to the other points. We selected $\sqrt{\varepsilon}$ such that each data point would be numerically connected with at least one other point in the graph. This value, which was found to be equal to 1000, turned out to make the graph of the data totally connected. The choice of this number was also coherent with the shape of the distribution of the distances (see Figure 2) in that, on average, each point is connected to a small fraction of the other points.

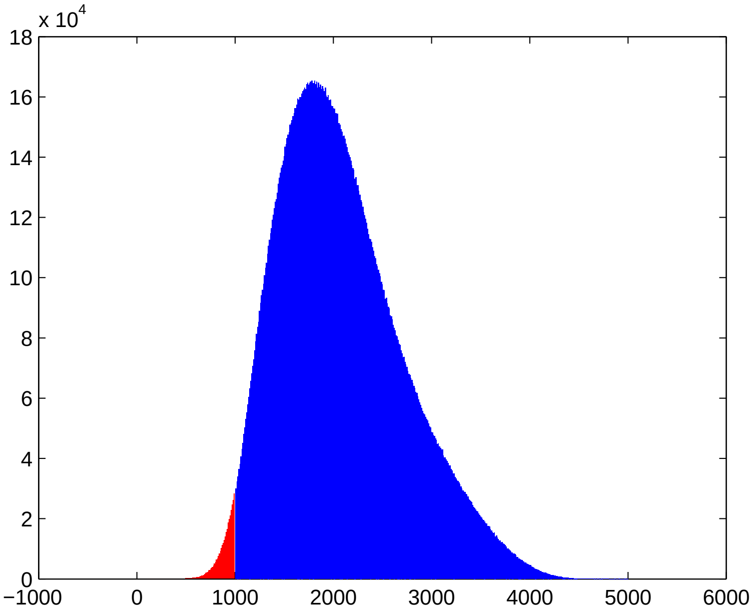


Fig. 2. The distribution of distances between all pairs of data points. The choice of the scale $\sqrt{\varepsilon} = 1000$ corresponds to having each data point connected to at least one other data point. The resulting graph happened to be totally connected. This histogram shows that the choice of this scale parameter leads to a sparse graph: each node is connected, on average, to a small number of other nodes.

We then renormalized the Gaussian weights using the Laplace-Beltrami normalization described in Section II-B. By doing so, our analysis focused on viewing the mouth as a constrained mechanical system. In order to obtain a low-dimensional parametrization of these nonlinear constraints, we computed the diffusion coordinates on this new graph. The spectrum of the diffusion matrix is plotted on Figure 3 and the embedding in the first 3 eigenfunctions is shown on Figure 4.

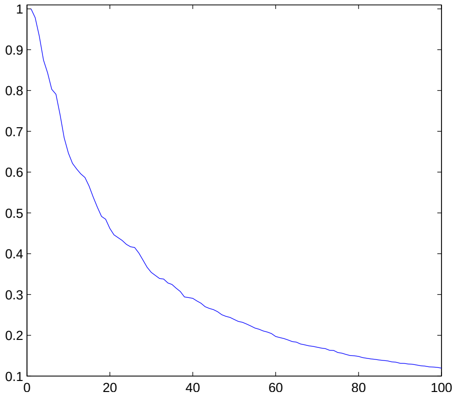


Fig. 3. The top 100 eigenvalues of the diffusion matrix for the lips data. The spectrum decays rapidly.

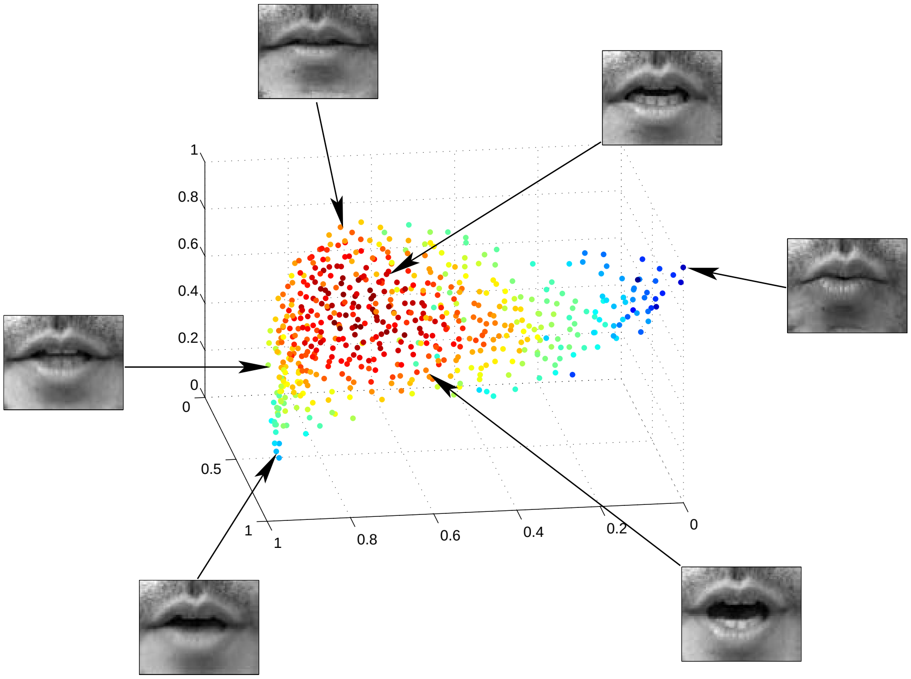


Fig. 4. The embedding of the lip data into the top 3 diffusion coordinates. These coordinates essentially capture two parameters: one controlling the opening of the mouth and the other measuring the portion of teeth that are visible.

The task we wanted to perform was isolated-word recognition on a small vocabulary. The example that we considered was that of identification of digits. Each word “zero”, “one”,..., “nine” is typically a sequence 25 to 40 frames that we need to project in the diffusion space[5]. In order to do so, we used the geometric harmonic extension scheme presented in Section II-C to extend each diffusion coordinate to the frames corresponding to the subject pronouncing the different digits. After this projection, each word can be viewed as a trajectory in the diffusion space. The word recognition problem now amounts to identifying trajectories in the diffusion space.

[5] Note that this second data set was *not* used to compute the diffusion maps.

We can now build a classifier based on comparing a new trajectory to a collection of labeled trajectories in a training set. We randomly selected 20 instances of each digit to form a training set, the remaining 20 being used as a testing set. In order to compare trajectories in the diffusion space, a metric is needed, and we chose to use the Hausdorff distance between two sets Γ_1 and Γ_2, defined as

$$d_H(\Gamma_1, \Gamma_2) = \max\left\{ \max_{x_2 \in \Gamma_2} \min_{x_1 \in \Gamma_1} \{\|x_1 - x_2\|\}, \max_{x_1 \in \Gamma_1} \min_{x_2 \in \Gamma_2} \{\|x_1 - x_2\|\} \right\} .$$

Although this distance does not use the temporal information, it has the advantage of not being sensitive to the choice of a parametrization or to the sampling density for either set Γ_1 and Γ_2. For a given trajectory Γ from the testing set, our classifier is a nearest-neighbor classifier for this metric, *i.e.*, the class of Γ is decided to be that of the nearest trajectory (for d_H) in the training set. The performance of this classifier averaged over 100 random trials is shown in Table I. In this case, the data set was embedded in 15 dimensions.

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.93**	0	0	0.01	0	0	0.06	0	0	0
one	0	**1**	0	0	0	0	0	0	0	0
two	0.05	0	**0.88**	0.05	0.01	0	0.01	0	0	0
three	0.01	0	0.02	**0.93**	0	0	0.01	0.01	0.01	0.01
four	0	0	0.01	0.01	**0.97**	0	0	0.01	0	0
five	0	0	0	0.01	0	**0.84**	0.01	0.14	0	0.01
six	0.04	0	0	0.01	0	0	**0.92**	0.02	0	0.01
seven	0.02	0	0	0.04	0	0.07	0.10	**0.69**	0.05	0.03
eight	0	0.01	0	0	0	0.03	0.01	0.04	**0.77**	0.14
nine	0	0	0	0.02	0	0	0	0.02	0.12	**0.85**

TABLE I

Classifier performance over 100 random trials. Each row corresponds the classification distribution of a given digit over then 10 classes. The data set was embedded in 15 dimensions.

The classification error ranges from 0% to 31% with an average of 12.2%. The best classification rate is achieved for the word “one” which, in terms of visual information, stands far away from the other digits. In particular, typical sequences of “one” involve frames with a round open mouth, with no teeth visible (see first row of Figure 5). These frames essentially never appear for other digits. The worst classification job is for the word “seven” which seems to be highly confused with the words “five” and “six”. As shown on Figure 5, typical instances of these words appear to be similar in that the central frames involve an open mouth with visible teeth. In the case of the “six” and “seven”, teeth from the lower jaws are visible because of the “s” sound. Regarding the similarity between “five” and “seven”, the ”f” and ”v” sounds translate into the lower lip touching the teeth of the upper jaw.

The accuracy that we obtain is comparable to former schemes [30], [41], while using significantly less preprocessing. For instance, in [30], the lips images are hand picked and stabilized using an affine motion model, while in [41] the contours of the lips are tracked by Active Contours. Our lips images are acquired by taping a continuous 5 minutes sequence and a simple cropping is performed to compensate for translations. We note that the above comparison is qualitative rather than quantitative, as the different schemes were applied to different datasets that are not publicly available.

B. Synchronization of head movement data

We now illustrate the concept of graph alignment as well as the algorithm presented in Section II-D. We recorded 3 movies of subjects wearing successively a yellow, red and black mask. Each subject was asked to move their head in front of the camcorder. We then considered the three sets consisting of all frames of each movie. Let YELLOW, RED and BLACK denote these sets. Our goal was to synchronize

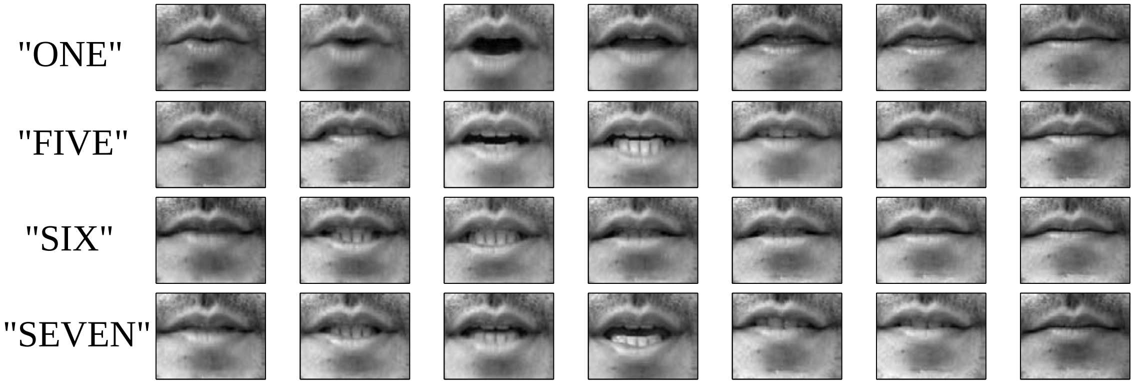


Fig. 5. Typical frames for the words “one”, “five”, “six”, “seven”.

the movements of the different masks by aligning the 3 diffusion embeddings. The objective of this experiment was twofold

- We first wanted to illustrate the importance of having a coordinate system capturing the intrinsic geometry of data sets. The intrinsic geometry is the basis of our alignment scheme: the key point is that, as we will show, all three sets exhibit approximately the same intrinsic geometry, and that the diffusion coordinates parameterize this geometry. It is to be noted that working directly in image space would be highly inefficient since any picture of the red or black mask is at a large distance from the set of pictures of the yellow mask (this is a straight consequence of the high dimensionality of the data). On the contrary, the diffusion coordinates will capture the intrinsic organization of each data sets, and therefore will provide a canonical representation of the sets that can be used for matching the data. Note also that our approach does not require any prior information on the type of data we are dealing with.
- The other point that we wished to illustrate is the importance of using the density-invariant diffusion maps. As we will show, although the three sets have approximately the same intrinsic geometry (the data points lie on the same 2D submanifold), the distribution of the points on this manifold are quite different. Therefore, it is necessary to employ the density re-normalization technique described in Section II-B.

These two points constitute the main ingredients for a successful alignment of the sets.

We now describe the experiment in more details. Each set of frames was regarded as a collection of points in $\mathbb{R}^{10000}$, where the dimensionality coincides with the number of pixels per image. Following the lines of our algorithm, we formed a graph from each set with Gaussian weights $\exp(-\|x_i - x_j\|^2/\varepsilon)$. The quantity $\|x_i - x_j\|$ represents the L^2 norm between images i and j, and here again, the scale was chosen so that each data point would be numerically connected to at least one other data point. We expect each set to lie approximately on a manifold of dimension 2, as each subject essentially moved their head along two angles α and β shown on Figure 6 and as the light conditions were kept the same during the recording. Therefore, each data sets is the expression of a highly constrained mechanical system, namely the articulation between the neck and the head.

It is clear that the density of points on this manifold is essentially arbitrary and varies with each subject and recording. Indeed, the density is essentially a function of the type of movement of each subject, their speed of execution, and also the type of mask that they were wearing. Since we were only interested in the space of constraints, that is the geometry of the manifold, we renormalized the Gaussian weights according to the algorithm described in Section II-B, and constructed a Markov chain that approximates the Laplace-Beltrami diffusion. Figure 7 shows the embedding in the first three eigenfunctions for each data set. They are extremely similar. We then defined 8 matching triplets of landmarks in each set. The landmarks were chosen to correspond to the main head positions. We computed the diffusion embedding

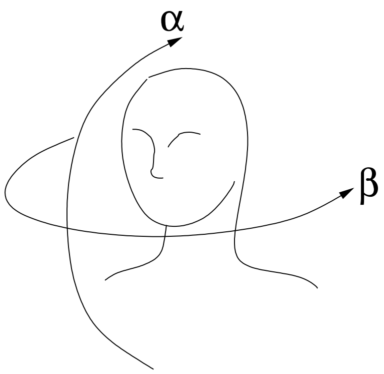
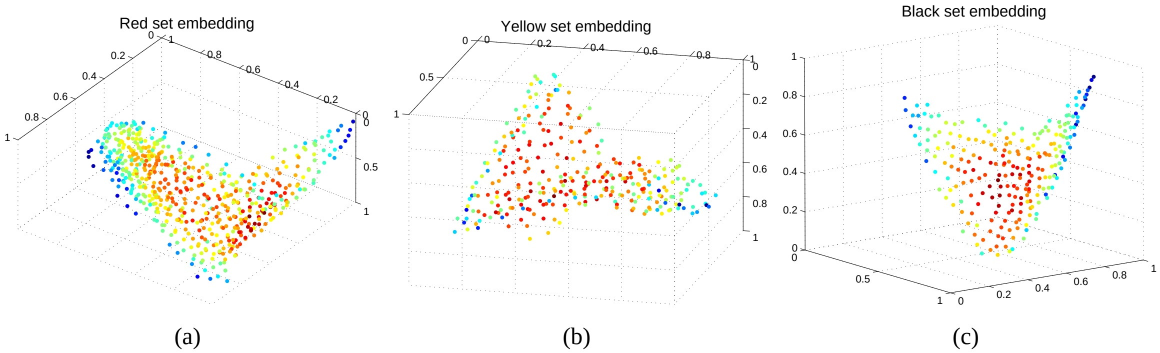


Fig. 6. Each subject essentially moved their head along the two angles α and β. There was almost no tilting of the head. Hence, the data points approximately lie on a submanifold of dimension 2.



(a)

(b)

(c)

Fig. 7. The embedding of each set in the first 3 diffusion coordinates. The color encodes the density of points. All three sets share this butterfly-shaped embedding

in 7 dimensions and we then calculated two affine functions $g_{YR} : \mathbb{R}^7 \longrightarrow \mathbb{R}^7$ and $g_{YB} : \mathbb{R}^7 \longrightarrow \mathbb{R}^7$ that match the landmarks from YELLOW to BLACK, and from YELLOW to RED.

Two conclusions can be drawn from this experiment. First, the diffusion embedding revealed that the data sets were approximately 2-dimensional, as expected (see Figure 7 for the embeddings in the first 3 diffusion coordinates). The diffusion coordinates captured the main parameters of variability, namely the angles α and β. From the embedding plots, it can be seen that all three embedded sets have strikingly similar shapes. *This supports our intuition that all sets should have similar intrinsic geometries*. From this observation, we were able to successfully compute two aligning functions g_{YB} and g_{YR}, and we used them to drive the movements of the black and red masks from those of the yellow mask. The result of the matching of the three data sets is shown on Figure 8. A live demo of this experiment can be found at [42].

The other conclusion concerns the importance of having used the density normalized diffusion coordinates. A key point in our analysis is that to compare the intrinsic geometries of each set, we need to be able to get rid of the influence of the points on the 2D submanifold. In order to underline the importance of this idea, we also computed the embedding of the three Yellow and BLACK without this renormalization. According to the discussion of Section II-B, the embedded sets should now reflect both the constraints (the intrinsic geometry) and the distribution of the points (the density on the submanifold). The result is shown on Figure 9, and although the embedding of the BLACK set still retain this butterfly shape that we previously obtained when renormalizing, the YELLOW set is now embedded as some portion of an ovoid. Although this statement can seem very qualitative, it is now clear that the alignment of these sets should fail. This experiment therefore underlines the importance of being able to compute density-invariant embeddings of the data.

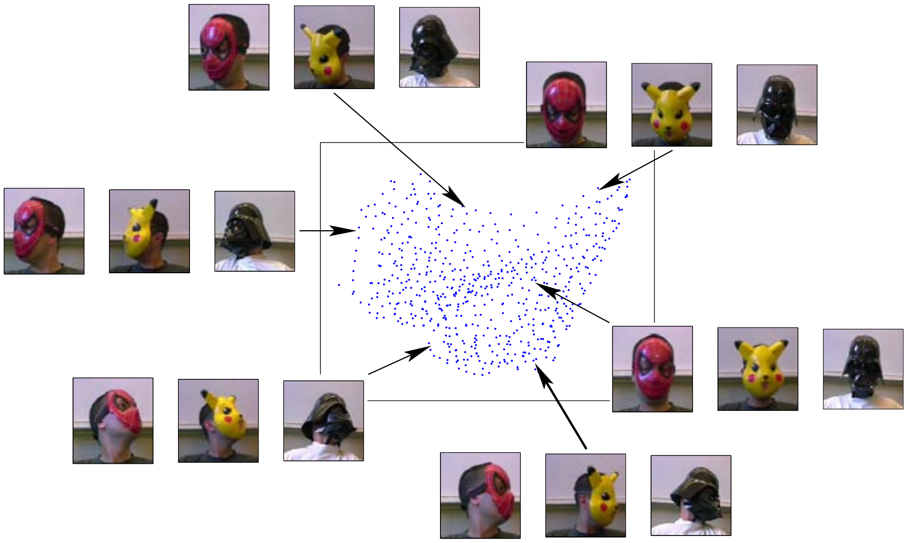
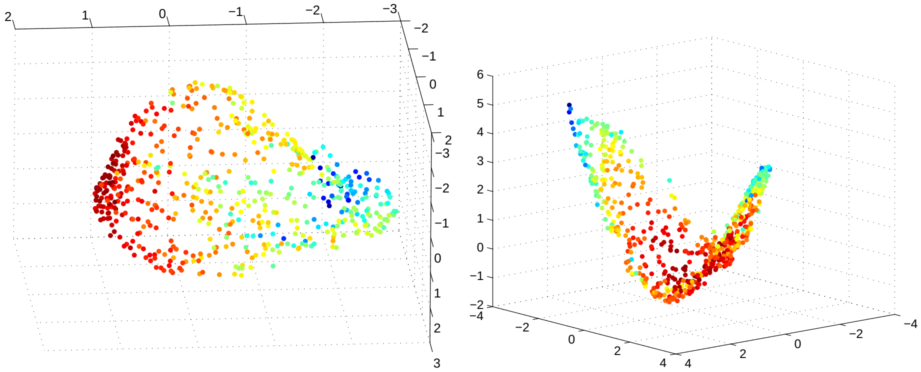


Fig. 8. The embedding of the YELLOW set in three diffusion coordinates and the various corresponding images after alignment of the RED and BLACK graphs to YELLOW.



(a)

(b)

Fig. 9. The embeddings of the YELLOW (a) and BLACK (b) sets in three diffusion coordinates without the density renormalization. These embedded sets now have very different shapes, and their alignment is impossible.

IV. Conclusion and future work

In this work we introduced diffusion techniques as a framework for data fusion and multi-cue data matching by addressing several key issues. First, we underlined the importance of the Laplace-Beltrami normalization for data fusion by showing that it allows to merge data sets produced by the same source but with different densities. In particular, the Laplace-Beltrami embedding provides a canonical, density-invariant embedding which is essential for data matching. Second, we suggested a new data fusion scheme, by extending spectral embeddings using the geometric harmonics framework. Finally, we presented a novel spectral graph alignment approach to data fusion.

Our scheme was successfully applied to lip-reading where we achieved high accuracy with minimal preprocessing. We also demonstrated the alignment of high-dimensional visual data ("rotating heads" sequence).

In the work presented, we have focused on the situation when all sources are highly correlated. In the future we plan on extending our approach to multi-cue data analysis by integrating different signals from weakly correlated sources into a unified representation. This should open the door to applications related to multi-sensor integration. Finally, we also are studying a spectral based approach to the analysis of signals as dynamical random processes. Our current work did not utilize the temporal information of the video sequences. By constructing a dynamical Markov process model, we intend to improve the lips reading accuracy.

V. Acknowledgments

The authors would like to thank Steven Zucker for helpful discussions during the course of this work and Andreas Glaser for helping us out for the data collection. We also express our thanks to the referees for their constructive questions and comments.

Appendix I
Existence and uniqueness of the stationary distribution

The goal of this section is to show that if the graph is connected, then the stationary distribution ϕ_0 is guaranteed to exist. The first step is to notice that the data set is finite, and therefore so is the state space of our Markov chain. Thus by a classical version of the Perron-Frobenius theorem, it suffices to prove that the chain is irreducible and aperiodic.

- The irreducibility is a mere consequence of the fact that the graph is connected. Indeed, let x_i and x_j be two data points, and let τ be the length of a path connecting x_i and x_j. Since the graph is connected, we know that $\tau < +\infty$. We conclude that $p_\tau(x_i, x_j) > 0$, which implies that the chain irreducible.
- Concerning the aperiodicity, remember that $w(\cdot, \cdot)$ represent the similarity between data points, so we can assume that for all data point x_i, we have $w(x_i, x_i) > 0$. Consequently, $p_1(x_i, x_i) > 0$, which implies that the chain is aperiodic.

Finally, we can conclude that our Markov chain has a unique stationary distribution ϕ_0.

Appendix II
Diffusion distance and eigenfunctions

The random walk constructed from a graph via the normalized graph Laplacian procedure yields a Markov matrix P with entries $p_1(x, y)$. As it is well known [15], this matrix is in fact conjugate to a symmetric matrix A with entries $a(x, y)$, given by

$$a(x,y) = \sqrt{\frac{d(x)}{d(y)}} p_1(x,y) = \frac{w(x,y)}{\sqrt{d(x)d(y)}} \, .$$

Therefore A has n eigenvalues $\lambda_0, ..., \lambda_{n-1}$ and orthonormal eigenvectors $v_0, ..., v_{n-1}$. In particular,

$$a(x,y) = \sum_{l=0}^{n-1} \lambda_l v_l(x) v_l(y) \, . \tag{9}$$

This implies that P has the same n eigenvalues. In addition, it has n left eigenvectors $\phi_0, ..., \phi_{n-1}$ and n right eigenvectors $\psi_0, ..., \psi_{n-1}$. Also, it can be checked that

$$\phi_l(y) = v_l(y) v_0(y) \text{ and } \psi_l(x) = v_l(x)/v_0(x) \, . \tag{10}$$

Furthermore, it can be verified that $v_0(x) = \sqrt{d(x)}/\sqrt{\sum_z d(z)}$, and therefore $\phi_0(y) = d(y)/\sum_z d(z)$ and $\psi_0(x) = 1$. In addition,

$$\phi_0(x)\psi_l(x) = \phi_l(x) \, . \tag{11}$$

It results from Equations 9 and 10 that P^t admits the following spectral decomposition:

$$p_t(x,y) = \sum_{l=0}^{n-1} \lambda_l^t \psi_l(x) \phi_l(y) \, , \tag{12}$$

together with the biorthogonality relation

$$\sum_{y\in\Omega} \phi_i(y)\psi_j(y) = \delta_{ij} \, , \tag{13}$$

where δ_{ij} is Kronecker symbol. Combining this last identity with Equation 11, one obtains

$$\sum_{y\in\Omega} \frac{\phi_i(y)\phi_j(y)}{\phi_0(y)} = \delta_{ij} \, .$$

This means that the system $\{\phi_l\}$ is orthonormal in $L^2(\Omega, 1/\phi_0)$. Therefore, if one fixes x, Equation 12 can interpreted as the decomposition of the function $p_t(x,\cdot)$ over this system, where the coefficients of decomposition are $\{\lambda_l^t \psi_l(x)\}$.

Now by definition,

$$D_t(x,z)^2 = \sum_{y\in\Omega} \frac{(p_t(x,y) - p_t(z,y))^2}{\phi_0(y)} = \|p_t(x,\cdot) - p_t(z,\cdot)\|^2_{L^2(\Omega,1/\phi_0)} \, .$$

Therefore,

$$D_t(x,y)^2 = \sum_{l=0}^{n-1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \, .$$

References

[1] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[2] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 6, no. 15, pp. 1373–1396, June 2003.

[3] D. Donoho and C. Grimes, "Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[4] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignement," Department of computer science and engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.

[5] R. Coifman and S. Lafon, "Diffusion maps," *Applied and Computational Harmonic Analysis*, 2006, to appear.

[6] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: a geometric framework for learning from examples," University of Chicago, Tech. Rep. TR-2004-06, 2004.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nyström method." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[8] Y. Bengio, J.-F. Paiement, and P. Vincent, "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," Université de Montréal, Tech. Rep. 1238, 2003.

[9] R. Coifman and S. Lafon, "Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions," *Applied and Computational Harmonic Analysis*, 2006, to appear.

[10] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[11] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonics analysis and structure definition of data: Multiscale methods," *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7432–7437, May 2005.

[12] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, "Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems," *Applied and Computational Harmonic Analysis*, 2006, to appear.

[13] S. Lafon and A. B. Lee, "Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning and data set parameterization," *IEEE Pattern Analysis and Machine Intelligence*, 2006.

[14] R. I. Kondor and J. D. Lafferty, "Diffusion kernels on graphs and other discrete input spaces," in *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, 2002, pp. 315–322.

[15] F. Chung, *Spectral graph theory*. CBMS-AMS, May 1997, no. 92.

[16] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Tran PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[17] Y. Weiss, "Segmentation using eigenvectors: A unifying view." in *ICCV*, 1999, pp. 975–982.

[18] M. Meila and J. Shi, "A random walk's view of spectral segmentation," *AI and Statistics (AISTATS)*, 2001.

[19] S. X. Yu and J. Shi, "Multiclass spectral clustering." in *Proc. IEEE Int. Conf. Computer Vision*, 2003, pp. 313–319.

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *Advances in Neural Information Processing*, 2001, pp. 585–591.

[21] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of markov chains," *The Annals of Applied Probability*, vol. 1, no. 1, pp. 36–61, 1991.

[22] M. Belkin and P. Niyogi, “Towards a theoretical foundation for laplacian-based manifold methods.” in *COLT*, 2005, pp. 486–500.

[23] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University, 1988.

[24] M. Gori, M. Maggini, and L. Sarti, “Exact and approximate graph matching using random walks.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 7, pp. 1100–1111, 2005.

[25] J. Ham, D. Lee, and L. Saul, “Semisupervised alignment of manifolds,” in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pp. 120–127.

[26] X. Bai, H. Yu, and E. R. Hancock, “Graph matching using spectral embedding and alignment.” in *ICPR (3)*, 2004, pp. 398–401.

[27] Y. Keselman, A. Shokoufandeh, M. F. Demirci, and S. J. Dickinson, “Many-to-many graph matching via metric embedding.” in *CVPR (1)*, 2003, pp. 850–857.

[28] A. W. Fitzgibbon, “Robust registration of 2d and 3d point sets,” in *Proceedings of the British Machine Vision Conference*, 2001, pp. 662–670.

[29] H. J. Wolfson and I. Rigoutsos, “Geometric hashing: An overview,” *IEEE Comput. Sci. Eng.*, vol. 4, no. 4, pp. 10–21, 1997.

[30] M. Aharon and R. Kimmel, “Representation analysis and synthesis of lip images using dimensionality reduction,” *Accepted to the International Journal of Computer Vision*.

[31] B. Christoph, C. Michele, and S. Malcolm, “Video rewrite: driving visual speech with audio,” in *SIGGRAPH ’97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 353–360.

[32] E. Cosatto and H. P. Graf, “Photo-realistic talking-heads from image samples.” *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 152–163, 2000.

[33] C. Bregler, S. Manke, and H. Hild, “Improving connected letter recognition by lipreading,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, 1993.

[34] N. Dettmer and M. Shah, “Visually recognizing speech using eigensequences,” *Computational Imaging and Vision*, pp. 345–371, 1997.

[35] I. Matthews, T. Cootes, A. Bangham, S. Cox, and R. Harvey, “Extraction of visual features for lipreading.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, 2002.

[36] A. V. Nefian, L. H. Liang, X. X. Liu, X. Pi, and K. Murphy, “Dynamic bayesian networks for audio-visual speech recognition,” *Journal of Applied Signal Processing,*, vol. 2002, no. 11,, pp. 1274–1288, 2002.

[37] J. Luettin, N. A. Thacker, and S. W. Beet, “Active shape models for visual speech feature extraction,” in *Speechreading by Humans and Machines*, ser. NATO ASI Series, Series F: Computer and Systems Sciences, D. G. Storck and M. E. H. (editors), Eds. Berlin: Springer Verlag, 1996, vol. 150, pp. 383–390.

[38] J. Luettin and N. A. Thacker, “Speechreading using probabilistic models,” *Computer Vision and Image Understanding*, vol. 65, no. 02, pp. 163–178, 1997.

[39] Y.-L. Tian, T. Kanade, and J. Cohn, “Robust lip tracking by combining shape, color and motion,” in *Proceedings of the 4th Asian Conference on Computer Vision (ACCV’00)*, January 2000.

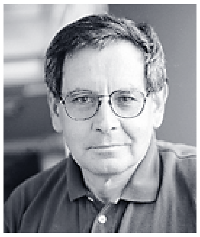
[40] I. Borg and P. Groenen, *Modern Multidimensional Scaling - Theory and Applications*. Springer-Verlag New York Inc., 1997.

[41] C. Bregler, S.Omohundro, M.Covell, M.Slaney, S.Ahmad, D.A.Forsyth, and J.A.Feldman, “Probabilistic models of verbal and body gestures,” in *Computer Vision in Man-Machine Interfaces*, R. Cipolla and A. eds, Eds. Cambridge University Press, 1998.

[42] S. Lafon, “Demo of the mask alignment,” 2005. [Online]. Available: http://www.math.yale.edu/∼sl349/demos_data/demos.htm.

Stéphane Lafon is a Software Engineer at Google. He received his B.Sc. degree in Computer Science from Ecole Polytechnique and his M.Sc. in Mathematics and Artificial Intelligence from Ecole Normale Supérieure de Cachan in France. He obtained his Ph.D. in Applied Mathematics at Yale University in 2004 and he was a research associate in the Mathematics Department during 2004-2005. He is currently with Google where his work focuses on the design, analysis and implementation of machine learning algorithms. His research interests are in data mining, machine learning and information retrieval.

Yosi Keller received the B.Sc. degree in electrical engineering in 1994 from The Technion-Israel Institute of Technology, Haifa. He received the M.Sc and Ph.D degree in electrical engineering from Tel-Aviv University, Tel-Aviv, in 1998 and 2003, respectively. From 1994 to 1998, he was an R&D Officer in the Israeli Intelligence Force. He is a visiting Assistant Professor with the Department of Mathematics, Yale University. His research interests include motion estimation and statistical pattern analysis.



Ronald R. Coifman is the Phillips Professor of Mathematics at Yale University. His research interests include: nonlinear Fourier analysis, wavelet theory, singular integrals, numerical analysis and scattering theory, and new mathematical tools for efficient computation and transcriptions of physical data, with applications to numerical analysis, feature extraction recognition and denoising. Professor Coifman, who earned his Ph.D. at the University of Geneva in 1965, is a member of the National Academy of Sciences and the American Academy of Arts and Sciences. He received the DARPA Sustained Excellence Award in 1996, the 1999 Pioneer Award from the International Society for Industrial and Applied Mathematics. He is a recipient of National Medal of Science.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Greedy Basis Pursuit

Patrick S. Huggins   Steven W. Zucker

Dept. of Computer Science, Yale University, P.O. Box 208285, New Haven, CT 06520

phone: (203) 432-1274;   fax: (203) 432-0593;   email: {huggins, zucker}@cs.yale.edu

Abstract

We introduce Greedy Basis Pursuit (GBP), a new algorithm for computing sparse signal representations using overcomplete dictionaries. GBP is rooted in computational geometry and exploits an equivalence between minimizing the ℓ^1-norm of the representation coefficients and determining the intersection of the signal with the convex hull of the dictionary. GBP unifies the different advantages of previous algorithms: like standard approaches to Basis Pursuit, GBP computes representations that have minimum ℓ^1-norm; like greedy algorithms such as Matching Pursuit, GBP builds up representations, sequentially selecting atoms. We describe the algorithm, demonstrate its performance, and provide code. Experiments show that GBP can provide a fast alternative to standard linear programming approaches to Basis Pursuit.

1 Introduction

The problem of computing sparse signal representations using an overcomplete dictionary arises in a wide range of signal processing applications [87, 34, 55], including image [10, 105], audio [68, 43], and video [6] compression and source localization [71]. The goal is to represent a given signal as a linear superposition of a small number of stored signals, called *atoms*, drawn from a larger set, called the *dictionary*. In traditional signal representation methods, such as the DCT or various wavelet transforms, the dictionary is simply a basis: the number of atoms in the dictionary is equal to the dimensionality of the signal space and representation is unique. By contrast, in an overcomplete dictionary the number of atoms is greater than the dimensionality of the signal space and representation is no longer unique; this enables flexibility in representation [72], ‘shiftability’ [93], and the use of multiple bases [62, 97], but it requires a criterion to select from among the (many) possibile representations. A natural one is sparsity, by which the representation selected is the one that uses as few atoms as possible.

Computing sparse representations is NP-hard [78, 31], and so several (heuristic) methods have been developed [72, 83, 19, 56]. These methods optimize various measures of sparsity, typically functions of the representation coefficients [66, 65], using, for example, greedy algorithms [72], gradient descent [69], linear programming [21], and global optimization [86]. Currently, the two most popular approaches are Matching Pursuit [72] and Basis Pursuit [20, 21].

Matching Pursuit (MP) is a greedy algorithm: a signal representation is iteratively built up by selecting the atom that maximally improves the representation at each iteration. While there is no guarantee that MP computes sparse representations, MP is easily implemented, converges quickly, and has good approximation properties [72, 100, 58]. Moreover, MP and one of its variants, Orthogonal Matching Pursuit (OMP) [83], can be shown to compute sparse (or nearly sparse) representations under some conditions [102, 58].

Basis Pursuit (BP), instead of seeking sparse representations directly, seeks representations that minimize the ℓ^1-norm of the coefficients. By equating signal representation with ℓ^1-norm minimization, BP reduces signal representation to linear programming [20, 21], which can be solved by standard methods [104]. Furthermore, BP methods can compute sparse solutions in situations where greedy algorithms fail [21]. Recent theoretical work shows that representations computed by BP are guaranteed to be sparse under certain conditions [37, 36, 51, 103].

While applying standard linear programming methods to compute minimum ℓ^1-norm signal representations is natural, such methods were developed with very different problems in mind and may not be ideally suited to the representation problem. For example, if the matrix corresponding to the dictionary is not sparse then the (normally fast) interior point methods advocated for BP [21] can be slow. Furthermore, the design required to produce examples on which greedy algorithms fail yet BP succeeds suggests that a greedy strategy could be successfully applied to minimum ℓ^1-norm representation.

In this article we develop a new algorithm for computing sparse signal representations, which we call Greedy Basis Pursuit (GBP). GBP is an algorithm for BP: it minimizes the ℓ^1-norm of the representation coefficients. However, unlike standard linear programming methods for BP, GBP proceeds much like MP, building up the representation by iteratively selecting atoms.

While algorithmically similar to MP, GBP differs from MP in two key ways: (1) GBP uses a novel criterion for selecting the next atom in the representation. The criterion is based on computational geometry, and effects a search for the intersection between the signal vector and the convex hull of the dictionary. (2) GBP may discard atoms that it has already selected; this is crucial, as it allows GBP to overcome the 'mistakes' that MP can make in atom selection when compared to BP [21].

While GBP returns the signal representation with the minimum ℓ^1-norm, and thus GBP enjoys the theoretical benefits of BP, the greedy strategy of GBP leads to computational gains when compared to standard linear programming methods. Experiments show our implementation of GBP to be faster than off-the-shelf linear programming packages on some signal representation problems, particularly high-dimensional problems with very overcomplete dictionaries.

The remainder of this paper is organized as follows. In Section 1.1 we formally state the sparse signal representation problem. In Section 2 we review current approaches to the problem. Section 3 provides the geometric interpretation of Basis Pursuit that underlies GBP. In Section 4 we describe the Greedy Basis Pursuit algorithm. Section 5 present the results of experiments with GBP. We discuss GBP in Section 6 and conclude in Section 7.

1.1 Problem Statement

Given a signal $\mathbf{x}$ and a dictionary $\mathcal{D}$ we seek a sparse representation of $\mathbf{x}$. We assume that $\mathbf{x}$ consists of d real valued measurements, that is, $\mathbf{x} \in \mathbb{R}^d$, for example, a sound wave sampled at d points. We assume that $\mathcal{D}$ consists of n atoms and is overcomplete, that is, $\mathcal{D} = \{\psi_i\}_{i=1}^n$ and $n > d$, and that the atoms are also d-dimensional and have unit norm, that is, $\forall \psi_i \in \mathcal{D}, \psi_i \in \mathbb{R}^d$ and $\|\psi_i\|_2 = 1$. A *representation* of $\mathbf{x}$ is a set of indices $\mathcal{I}$ into $\mathcal{D}$, where $\mathcal{I} \subseteq \{1, \ldots, n\}$, and a corresponding set of coefficients $\mathcal{A} = \{\alpha_i\}_{i \in \mathcal{I}}$ such that

$$\mathbf{x} = \sum_{i \in \mathcal{I}} \alpha_i \psi_i \tag{1}$$

A representation is *sparse* if the number of atoms used, $|\mathcal{I}|$ (here $|\cdot|$ denotes cardinality), is minimized over all possible representations.

Equivalently, in matrix notation, given a (column) vector $\mathbf{x} \in \mathbb{R}^d$ corresponding to the signal, and a $d \times n$ matrix $\mathbf{D}$ corresponding to the dictionary, where the ith column of $\mathbf{D}$ is the atom ψ_i, the sparse signal representation problem is then to compute a (column) vector $\alpha \in \mathbb{R}^n$ solving

$$\text{Minimize } \ \|\alpha\|_0 \quad \text{subject to } \ \mathbf{D}\alpha = \mathbf{x} \tag{2}$$

where $\|\alpha\|_0$ is the ℓ^0-norm of α, defined to be the number of nonzero entries of α. In general, the equality constraint can be relaxed to give a corresponding approximation problem; see [78, 100, 102, 103].

BP replaces the ℓ^0-norm with the ℓ^1-norm, seeking representations that minimize $\sum_{i \in I} |\alpha_i|$. In matrix form this corresponds to

$$\text{Minimize } \ \|\alpha\|_1 \quad \text{subject to } \ \mathbf{D}\alpha = \mathbf{x} \tag{3}$$

GBP solves 3. The approximation problem corresponding to BP is called Basis Pursuit Denoising; see [21, 69, 42].

2 Related work

The design of GBP draws on previous work in sparse signal representation, particularly the contrast between MP and BP, and on ideas from subset selection, which we summarize here. We also highlight some unexplored connections between sparse signal representation and linear programming.

2.1 Matching Pursuit

Matching Pursuit (MP) [72] is the prototypical greedy algorithm [23] applied to sparse signal representation. MP is currently the most popular algorithm for computing sparse signal representations using an overcomplete dictionary, and is used in a variety of applications [10, 84, 6]. MP has also spawned several variants [46, 63, 47], including Orthogonal Matching Pursuit (OMP) [83, 32], which itself has several variants [54, 24, 88].

MP computes a signal representation by greedily constructing a sequence of approximations to the signal, $\tilde{\mathbf{x}}^{(0)}, \tilde{\mathbf{x}}^{(1)}, \tilde{\mathbf{x}}^{(2)}, \ldots$, where each consecutive approximation is closer to the signal. MP begins with an 'empty' representation, $\tilde{\mathbf{x}}^{(0)} = 0$, and at each iteration augments the current representation by selecting the atom from the dictionary which is closest to the residual, $\tilde{\mathbf{x}}^{(t+1)} = \tilde{\mathbf{x}}^{(t)} + \alpha^{(t)}\psi^{(t)}$, where $\psi^{(t)} = \arg\max_{\psi \in \mathcal{D}} \langle \psi, \mathbf{x} - \tilde{\mathbf{x}} \rangle$.

MP is easy to implement, has a guaranteed exponential rate of convergence [72, 100, 58], and recovers relatively sparse solutions [102], particularly compared to earlier approaches such as the Method-of-Frames [29, 21].

A drawback of MP applied to sparse representation is its greediness. It is possible to construct signal representation problems where, because of its greediness, MP (or OMP) intially selects an atom that is not part of the optimal sparse representation; as a result, many of the subsequent atoms selected by MP simply compensate for the poor initial selection [33, 21]. This shortcoming motivated the development of BP, which succeeds on these problems[21]; recent theoretical work explains this phenomenon [37, 36, 51].

These problems are also motivation for the development of GBP. Here MP fails because of its poor intial selection of atoms; however, the atoms intially selected by MP are not necessarily bad in general, after all, these problems are specially designed for MP to fail on. For MP to succeed on these problems, it would need to either make 'better' atom selections or be able to discard 'bad' atoms to recover from poor selections (or both). GBP adapts the greedy strategy to incorporate both of these ideas and compute the same representations as BP.

2.2 Basis Pursuit

Basis Pursuit (BP) [19, 20, 21] approaches sparse signal representation by changing the problem to one of minimizing the ℓ^1-norm of the representation coefficients. This can be interpreted as assuming a 'sparse prior' on the representation coefficients [80]. The ℓ^1-norm in particular implies that the resulting representations are sparse in the ℓ^0-norm sense under certain conditions [37, 36, 51] and algorithmically equates sparse signal representation with linear programming.

A linear program is defined as follows: Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, a (column) vector $\mathbf{b} \in \mathbb{R}^m$, and a (column) vector $\mathbf{c} \in \mathbb{R}^n$, compute a (column) vector $\mathbf{x} \in \mathbb{R}^n$ satisfying

$$\text{Minimize } \mathbf{c}^T\mathbf{x} \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{b},\ x_i \geq 0 \tag{4}$$

The signal representation problem is posed in BP as a linear program with the following assignments (the variables on the right hand side are as defined in Section 1.1 and the variables on the left hand side plug into the linear program above):

$$\begin{aligned} \mathbf{A} &\leftarrow [\,\psi_1\ \psi_2\ \cdots\ \psi_n\ -\psi_1\ -\psi_2\ \cdots\ -\psi_n\,] \\ \mathbf{b} &\leftarrow \mathbf{x} \\ \mathbf{c} &\leftarrow [\,1\ 1\ \cdots\ 1\,] \\ \mathbf{x} &\leftarrow [\,\alpha_1\ \alpha_2\ \cdots\ \alpha_n\,] \end{aligned}$$

Minimizing $\mathbf{c}^T\mathbf{x}$ is equivalent to minimizing the ℓ^1-norm of the coefficients. Note that $\mathbf{A}$, corresponding to the dictionary, is doubled to include the negative of each atom; this is due to the linear programming constraint that the coefficients be nonnegative 4.

Chen *et al.* [20, 21] describe two algorithms for BP, BP-Simplex and BP-Interior, which are the well-known simplex and interior point methods of linear programming [104] applied to signal representation. The choice of which BP algorithm to use depends on the structure of the dictionary: for dictionaries that have fast transforms, BP-Interior exploits these transforms in the solution of the corresponding linear program. However, the running time of linear programming is still typically an order of magnitude slower than that of MP on realistic problems [21].

While standard linear programming methods have been highly tuned over time, they are not necessarily ideally suited to the specific problem of computing signal representations. For example, many linear programming methods assume that the matrix $\mathbf{A}$ is sparse, as is the case for constraints that arise in typical operations research problems, while this may not be the case in signal representation problems. This raises the possibility that alternative approaches could prove more efficient for the particular problem of signal representation. Some inspiration for an alternative approach is provided by Chen *et al.* [21], who contrast MP and BP-Simplex, characterizing MP as a 'build-up' approach and BP-Simplex as a 'swap-down' approach. If **A** is not sparse, then the swaps (or pivots) executed by BP-Simplex can be costly, in the computation of an individual swap, in the number of swaps, and in the computation of an initial basis. GBP instead takes the 'build-up' approach to solving linear programming.

2.3 Subset selection

Sparse signal representation is closely related to the problem of subset selection for regression, i.e., determining the optimal subset of variables on which to regress a data set [74]. In sparse signal representation, the signal corresponds to the data set, while the atoms correspond to the variables. In fact, MP was inspired by Projection Pursuit [50, 61], in particular its use as a regression algorithm [49]. Given this connection, it should not be surprising that some algorithmic ideas in sparse signal representation correspond to earlier work in regression. For example, in Forward Selection the optimal subset is constructed by starting with the empty subset and iteratively adding variables to it, selecting at each iteration the variable that accounts for most of the residual variance; this is essentially what OMP does. Backward Elimination, which starts with the full set of variables and iteratively pares it down, has similarly been adapted for signal representation [59, 25].

One standard algorithm for subset selection in regression which appears to have no analogue in sparse signal representation is Efroymson's algorithm [41], also called step-wise regression, proceeds like Forward Selection, but, like Backward Elimination, drops variables from the subset as they become irrelevant. GBP follows a similar strategy, iteratively selecting atoms and occasionally discarding them.

2.4 Linear programming

While Basis Pursuit represents the first formal casting of signal representation as linear programming, linear programming has long been used in sparse signal representation, particularly for deconvolution in various applications [40, 8, 79]. It is therefore not surprising that developments in sparse signal representation closely parallel earlier developments in linear programming.

Examining the literature in linear programming reveals that MP and OMP have linear programming analogues: MP is technically equivalent to one of the earliest (1948) methods developed for linear programming, called von Neumann's algorithm [26]. Similarly, OMP is equivalent to a phase I algorithm [67] for the simplex method.

GBP builds up a solution to a linear programming problem; several linear programming methods adopt a similar strategy, solving increasingly complex problems as constraints or variables are iteratively introduced [98, 92, 81]; see also [60]. We remark that one method, an interior point method called the gravitational method [77, 18], can be shown to be equivalent to GBP when applied to the problem dual to (4). Empirically, the gravitational method is faster than standard methods on some problems [18], which is consistent with our results.

3 The Geometry of Basis Pursuit

GBP is based on computatonal geometry, specifically on the following geometric interpretation of BP. Given a signal $\mathbf{x}$ and a dictionary $\mathcal{D}$, let $\mathbf{conv}(\mathcal{D})$ denote the convex hull of $\mathcal{D}$; *the vertices of the facet of* $\boldsymbol{conv}(\mathcal{D})$ *intersected by the vector* $\mathbf{x}$ *are the atoms in the minimum* ℓ^1*-norm representation of* $\mathbf{x}$.

To see this, treat the signal as a vector and the atoms as points in $\mathbb{R}^d$. First consider the set of signals that have representations α such that $\|\alpha\|_1 = 1$. By definition, this is the convex hull of the dictionary

$$\mathbf{conv}(\mathcal{D}) = \left\{ \mathbf{x} \;\middle|\; \mathbf{x} = \sum_{i \in \mathcal{I}} \alpha_i \psi_i \;\text{ and }\; \sum_{i \in \mathcal{I}} \alpha_i = 1,\; \alpha_i > 0 \right\}$$

Note that because $\|\psi_i\|_2 = 1$, $\mathbf{conv}(\mathcal{D})$ is a polytope inscribed in the unit sphere. Let $\mathbf{x}_{\mathcal{D}}$ be the point of intersection between the vector $\mathbf{x}$ and the boundary of $\mathbf{conv}(\mathcal{D})$. $\mathbf{x}_{\mathcal{D}}$ lies on the boundary of $\mathbf{conv}(\mathcal{D})$ and can be represented as a linear combination of the vertices of the facet of $\mathbf{conv}(\mathcal{D})$ containing $\mathbf{x}_{\mathcal{D}}$; call this facet $F_{\mathbf{x}}$. This representation is the minimum ℓ^1-norm representation: its ℓ^1-norm is 1, and it is impossible to construct a representation with ℓ^1-norm less than 1. The

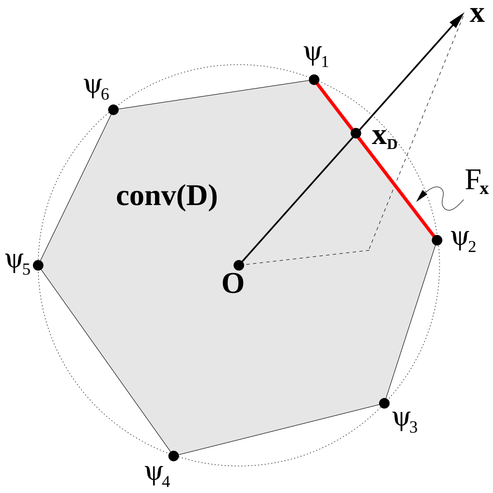


Figure 1: *A geometric interpretation of Basis Pursuit.* The signal vector $\mathbf{x}$ intersects the facet $F_{\mathbf{x}}$ of the convex hull of the dictionary, shown in gray. The vertices of $F_{\mathbf{x}}$, ψ_1 and ψ_2, are the atoms in the Basis Pursuit representation of $\mathbf{x}$.

minimum ℓ^1-norm representation of $\mathbf{x}$ is simply a scaling of the minimum ℓ^1-norm representation of $\mathbf{x}_{\mathcal{D}}$, and the atoms in the representation are the same. See Figure 1. (Note that if we know the atoms in a representation of $\mathbf{x}$ it is straightforward to calculate the corresponding coefficients.)

Thus BP is equivalent to finding the facet of $\mathbf{conv}(\mathcal{D})$ which intersects $\mathbf{x}$. Computing this intersection is known to reduce to linear programming [90]; to our knowledge, the converse is known [15] but never utilized to solve linear programming. We use this equivalence to drive GBP.

A previous geometric interpretation of sparse representation [14] recognizes that in two dimensions BP computes representations with atoms that ‘enclose’ $\mathbf{x}$. The intepretation provided here can be viewed as the generalization of this notion to higher dimensions.

4 The Greedy Basis Pursuit Algorithm

Given the equivalence between BP and finding the facet of the convex hull of the dictionary that intersects the signal vector, we propose Greedy Basis Pursuit (GBP). GBP computes the minimum ℓ^1-norm representation by searching for this facet directly.

The main idea behind GBP is to find the facet of interest by iteratively 'pushing' a hyperplane onto the surface of the convex hull of the dictionary until it coincides with the supporting hyperplane containing the facet. This approach is inspired by gift-wrapping methods [17, 64, 99] for the convex hull problem in computational geometry [91]. To adapt gift-wrapping to the problem of finding a particular facet, we need to specify how the initial hyperplane is chosen and the direction in which the 'wrapping' proceeds at each iteration. Below we describe the GBP algorithm, prove its convergence, and discuss implementation issues.

4.1 The main algorithm

GBP takes as input a signal $\mathbf{x} \in \mathbb{R}^d$ and an overcomplete dictionary $\mathcal{D} = \{\psi_i\}_{i=1}^n$, where $n > d$ and $\forall i$, $\psi_i \in \mathbb{R}^d$ and $\|\psi_i\|_2 = 1$, and outputs a representation of $\mathbf{x}$ as a set of indices $\mathcal{I} \subseteq \{1, \ldots, n\}$ and a corresponding set of coefficients $\mathcal{A} = \{\alpha_i\}_{i \in \mathcal{I}}$ such that $\mathbf{x} = \sum_{i \in \mathcal{I}} \alpha_i \psi_i$. Note we assume that if $\psi_i \in \mathcal{D}$ then $-\psi_i \in \mathcal{D}$; see section 2.2.

GBP greedily searches for the facet of $\mathbf{conv}(\mathcal{D})$ that intersects $\mathbf{x}$, call it $F_{\mathbf{x}}$. GBP proceeds by iteratively constructing a sequence of hyperplanes, $H^{(0)}, H^{(1)}, H^{(2)}, \ldots$, supporting $\mathbf{conv}(\mathcal{D})$. (We use the superscript (t) to denote iteration t.) At each iteration, GBP maintains a set of indices $\mathcal{I}^{(t)}$ and a set of coefficients $\mathcal{A}^{(t)}$, defining an approximation to $\mathbf{x}$: $\tilde{\mathbf{x}}^{(t)} = \sum_{i \in \mathcal{I}^{(t)}} \alpha_i \psi_i$, and a normal vector $\mathbf{n}^{(t)}$. The current hyperplane $H^{(t)}$ is defined to have normal $\mathbf{n}^{(t)}$ and contain the set $\{\psi_i\}_{i \in \mathcal{I}^{(t)}}$. Each consecutive hyperplane $H^{(t+1)}$ is a rotation of the current hyperplane $H^{(t)}$ determined by $\tilde{\mathbf{x}}^{(t)}$. GBP stops when $H^{(t)}$ contains $F_{\mathbf{x}}$ (and therefore $\tilde{\mathbf{x}}^{(t)} = \mathbf{x}$).

4.1.1 Initialization

As we do not *a priori* know the orientation of $F_{\mathbf{x}}$, we optimistically choose the initial supporting hyperplane $H^{(0)}$ to have normal $\mathbf{n}^{(0)} = \mathbf{x}/\|\mathbf{x}\|_2$. In general $H^{(0)}$ will intersect only one vertex of $\mathbf{conv}(\mathcal{D})$, in particular the atom ψ_{i_0}, where $i_0 = \arg\max_i \langle \psi_i, \mathbf{n}^{(0)} \rangle$. To see this, consider a hyperplane with normal $\mathbf{n}^{(0)}$ at some distance greater than 1 away from the origin; if we move this hyperplane in the negative normal direction (towards the origin), the first point of $\mathbf{conv}(\mathcal{D})$ it will intersect is ψ_{i_0}. (Note that this is also the first atom selected by MP and OMP.) Thus $\mathcal{I}^{(0)} = \{i_0\}$;

this gives us $\alpha_{i_0} = \langle \psi_{i_0}, \mathbf{x} \rangle$, $\mathcal{A}^{(0)} = \{\alpha_{i_0}\}$, and $\tilde{\mathbf{x}}^{(t)} = \alpha_{i_0}\psi_{i_0}$. For convenience, we denote the set of currently selected atoms by $\Psi^{(t)} = \{\psi_i\}_{i \in \mathcal{I}^{(t)}}$.

4.1.2 Iteration

Each consecutive hyperplane $H^{(t+1)}$ is constructed by rotating $H^{(t)}$ in a 2-dimensional plane around a pivot point until another vertex of $\mathbf{conv}(\mathcal{D})$ is intersected. The plane of rotation and the pivot point are defined in terms of $\tilde{\mathbf{x}}^{(t)}$. We define $\tilde{\mathbf{x}}^{(t)}$ to be the best current approximation to $\mathbf{x}$ using $\Psi^{(t)}$ and positive coefficients, that is, $\tilde{\mathbf{x}}^{(t)} = \sum_{i \in \mathcal{I}^{(t)}} \alpha_i \psi_i$, where $\alpha_i > 0$ and $\|\tilde{\mathbf{x}}^{(t)} - \mathbf{x}\|_2$ is minimized. Note that $\tilde{\mathbf{x}}^{(t)}$ is the projection of $\mathbf{x}$ onto the convex cone spanned by $\Psi^{(t)}$ with the origin at the apex; we provide details on computing $\tilde{\mathbf{x}}^{(t)}$ in Section 4.1.3. Let $\tilde{\mathbf{x}}_H^{(t)}$ be the intersection of the vector $\tilde{\mathbf{x}}^{(t)}$ with $H^{(t)}$. If $d_H^{(t)}$ is the (orthogonal) distance from the hyperplane to the origin, i.e., $d_H^{(t)} = \langle \psi_i, \mathbf{n} \rangle, \forall i \in \mathcal{I}^{(t)}$, then

$$\tilde{\mathbf{x}}_H^{(t)} = \left(d_H^{(t)} / \langle \tilde{\mathbf{x}}^{(t)}, \mathbf{n}^{(t)} \rangle \right) \tilde{\mathbf{x}} \tag{5}$$

Let $\mathbf{r}^{(t)}$ denote the residual vector, $\mathbf{r}^{(t)} = \mathbf{x} - \tilde{\mathbf{x}}^{(t)}$. Define $\mathbf{v}^{(t)}$ to be the unit vector in the direction of $\mathbf{r}^{(t)}$ projected onto $H^{(t)}$.

$$\mathbf{v}^{(t)} = \frac{\mathbf{r}^{(t)} - \langle \mathbf{r}^{(t)}, \mathbf{n}^{(t)} \rangle \mathbf{n}^{(t)}}{\|\mathbf{r}^{(t)} - \langle \mathbf{r}^{(t)}, \mathbf{n}^{(t)} \rangle \mathbf{n}^{(t)}\|} \tag{6}$$

The plane of rotation is the 2-dimensional plane defined by the point $\tilde{\mathbf{x}}_H^{(t)}$ and the vectors $\mathbf{n}^{(t)}$ and $\mathbf{v}^{(t)}$. The pivot point around which H is rotated is $\tilde{\mathbf{x}}_H^{(t)}$.

To compute the first vertex which the hyperplane intersects under this rotation, we order the atoms by the angle θ they form with $\mathbf{v}$, where θ is given by

$$\theta_i = \arctan(\langle \psi_i - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{n}^{(t)} \rangle / \langle \psi_i - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{v}^{(t)} \rangle)$$

The atom selected is then ψ_k where

$$k = \arg\min_i \theta_i \tag{7}$$

Once selected, the atom ψ_k is added to the set $\Psi^{(t)}$ and a new approximation to $\mathbf{x}$ is computed, $\tilde{\mathbf{x}}^{(t+1)}$. In this new approximation, some atoms in $\Psi^{(t)} \cup \{\psi_k\}$ may become extraneous: they are discarded to form $\Psi^{(t+1)}$; see Section 4.1.3 below.

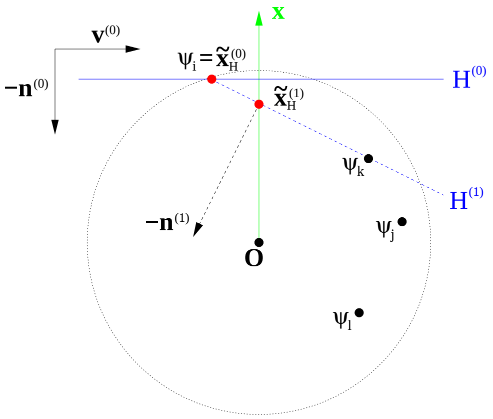


Figure 2: A schematic of the first iteration of GBP. The intial hyperplane $H^{(0)}$ has normal $\mathbf{n}^{(0)}$ in the direction of the signal $\mathbf{x}$ and contains ψ_i. The atoms are projected from $\mathbb{R}^d$ to the $\mathbf{n}^{(0)}$-$\mathbf{v}^{(0)}$ plane (shown) and sorted by θ. The second atom selected is ψ_k, corresponding to a rotation of $H^{(0)}$ around $\tilde{\mathbf{x}}_H^{(0)}$ to $H^{(1)}$. Note that $\mathbf{v}^{(1)}$ is orthogonal to the $\mathbf{n}^{(0)}$-$\mathbf{v}^{(0)}$ plane (and therefore is not shown).

The new hyperplane $H^{(t+1)}$ can now be computed; it has normal

$$\mathbf{n}^{(t+1)} = \frac{-\langle \psi_k - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{n}^{(t)} \rangle \mathbf{v}^{(t)} + \langle \psi_k - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{v}^{(t)} \rangle \mathbf{n}^{(t)}}{\| - \langle \psi_k - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{n}^{(t)} \rangle \mathbf{v}^{(t)} + \langle \psi_k - \tilde{\mathbf{x}}_H^{(t)}, \mathbf{v}^{(t)} \rangle \mathbf{n}^{(t)} \|} \tag{8}$$

and contains $\tilde{\mathbf{x}}_H^{(t+1)}$.

The procedure is repeated until $\tilde{\mathbf{x}}^{(t)} = \mathbf{x}$, that is, $H^{(t)}$ contains $F_{\mathbf{x}}$.

Figure 4.1.2 provides a visualization of GBP in action in three dimensions.

4.1.3 Computational details

At each iteration we compute $\tilde{\mathbf{x}}^{(t)}$ as the projection of $\mathbf{x}$ onto the convex cone of $\Psi^{(t)}$.

One approach to computing this projection is to maintain an orthogonal basis for the span of $\Psi^{(t)}$, updating it as atoms are added to $\Psi^{(t)}$, as in OMP [83]; this is impractical in our case as most iterative orthogonalization procedures are order-dependent and hyperplane rotation may cause us to discard arbitrary atoms from $\Psi^{(t)}$.

Instead we maintain a biorthogonal system consisting of $\Psi^{(t)}$ and $\tilde{\Psi}^{\perp(t)}$, the set of vectors

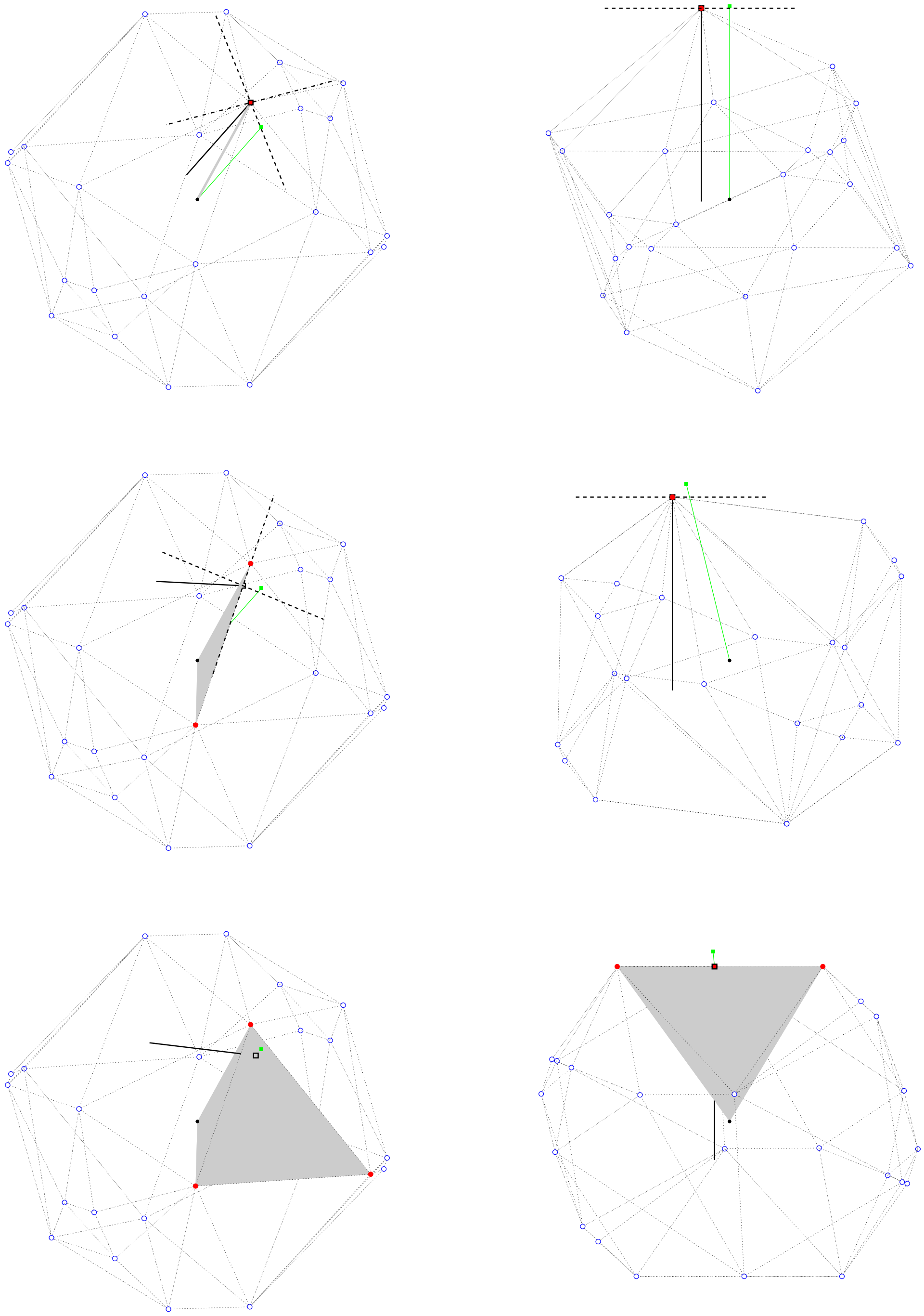


Figure 3: GBP in action on a 3-dimensional problem. Each row depicts one iteration, the left column from a fixed viewpoint, the right column projected to the $\mathbf{n}^{(t)}$-$\mathbf{v}^{(t)}$ plane. The signal vector is green, the unselected atoms blue circles, the selected atoms red discs, the convex cone of $\Psi^{(t)}$ is gray, the normal is the solid black line, and two vectors in $H^{(t)}$ are the dashed lines.

Algorithm 1 Greedy Basis Pursuit

```
Input
  • A signal $\mathbf{x} \in \mathbb{R}^d$
  • A dictionary $\mathcal{D} = \{\psi_i\}_{i=1}^n$,
  • A threshold $\epsilon \geq 0$
Output
A representation of $\mathbf{x}$, consisting of
  • A set of indices $\mathcal{I} \subseteq \{1, \ldots, n\}$
  • A set of coefficients $\mathcal{A} = \{\alpha_i\}_{i \in \mathcal{I}}$

such that $\mathbf{x} - \sum_{i \in \mathcal{I}} \alpha_i \psi_i < \epsilon$
Procedure
  1. Initialize
     (a) Select the first atom
         $k \leftarrow \arg\max_{i \in \{1,\ldots,n\}} \langle \mathbf{x}, \psi_i \rangle$
     (b) Compute the initial approximation
         $\alpha_k \leftarrow \langle \mathbf{x}, \psi_k \rangle, \quad \mathcal{I}^{(0)} \leftarrow \{k\}, \quad \mathcal{A}^{(0)} \leftarrow \{\alpha_k\}$
     (c) Initialize the biorthogonal system
         $\tilde{\Psi}^{\perp} \leftarrow \{\psi_k\}$
     (d) Initialize the hyperplane
         $\tilde{\mathbf{x}}^{(0)} \leftarrow \alpha_k \psi_k, \quad \mathbf{n} \leftarrow \mathbf{x}/\|\mathbf{x}\|, \quad \mathbf{r} \leftarrow \mathbf{x} - \tilde{\mathbf{x}}$
  2. Repeat until $\|\mathbf{r}\| < \epsilon$
     (a) Compute the center and plane of rotation
         $\tilde{\mathbf{x}}_H \leftarrow (\langle \psi_i, \mathbf{n} \rangle / \langle \tilde{\mathbf{x}}, \mathbf{n} \rangle)\, \tilde{\mathbf{x}}$, for any $i \in \mathcal{I}$
         $\mathbf{v} \leftarrow (\mathbf{r} - \langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n}) / \|\mathbf{r} - \langle \mathbf{r}, \mathbf{n} \rangle \mathbf{n}\|$
     (b) Project atoms into the n-v-plane and select the next atom
         $k \leftarrow \arg\min_{i, \in \{1,\ldots,n\}} \tan^{-1} \frac{\langle \psi_i - \tilde{\mathbf{x}}_H, \mathbf{n} \rangle}{\langle \psi_i - \tilde{\mathbf{x}}_H, \mathbf{v} \rangle}$
     (c) Compute the new representation and update the biorthogonal system
         $\{\mathcal{I}, \mathcal{A}, \tilde{\Psi}^{\perp}\} \leftarrow \mathrm{AddAtom}(\mathbf{x}, \mathcal{I}, \mathcal{A}, \psi_k, \tilde{\Psi}^{\perp})$
     (d) Discard any extraneous atoms
         while $\exists \alpha_i \leq 0, i \in \mathcal{I}$ do
         $\{\mathcal{I}, \mathcal{A}, \tilde{\Psi}^{\perp}\} \leftarrow \mathrm{SubtractAtom}(\mathbf{x}, \mathcal{I}, \mathcal{A}, \psi_j, \tilde{\Psi}^{\perp})$
     (e) Update the hyperplane parameters
         $\tilde{\mathbf{x}} \leftarrow \sum_{i \in \mathcal{I}} \alpha_i \psi_i$
         $\mathbf{n} \leftarrow \frac{-\langle \psi_k - \tilde{\mathbf{x}}_H, \mathbf{n} \rangle \mathbf{v} + \langle \psi_k - \tilde{\mathbf{x}}_H, \mathbf{v} \rangle \mathbf{n}}{\|-\langle \psi_k - \tilde{\mathbf{x}}_H, \mathbf{n} \rangle \mathbf{v} + \langle \psi_k - \tilde{\mathbf{x}}_H, \mathbf{v} \rangle \mathbf{n}\|}$
         $\mathbf{r} \leftarrow \mathbf{x} - \tilde{\mathbf{x}}$
```

biorthogonal to $\Psi^{(t)}$. Each element $\tilde{\psi}_i^{\perp(t)}$ of $\tilde{\Psi}^{\perp(t)}$ satisfies the following two equations:

$$\langle \psi_i, \tilde{\psi}_i^{\perp(t)} \rangle = 1 \tag{9}$$

$$\langle \psi_i, \tilde{\psi}_j^{\perp(t)} \rangle = 0, \text{ if } i \neq j \tag{10}$$

The biorthogonal vector $\tilde{\psi}_i^{\perp(t)}$ can be understood as the component of $\psi_i^{(t)}$ that is orthogonal to all of the other vectors in $\Psi^{(t)}$, appropriately scaled. That is, if we express an atom $\psi_i \in \Psi^{(t)}$ as

$$\psi_i = \psi_i^{\|(t)} + \psi_i^{\perp(t)} \tag{11}$$

where $\psi_i^{\|(t)}$ is the component of ψ_i lying in the span of $\Psi^{(t)} - \{\psi_i\}$

$$\psi_i^{\|(t)} = \sum_{j \in \mathcal{I}^{(t)}, j \neq i} \beta_{ij}^{(t)} \psi_j \tag{12}$$

and $\psi_i^{\perp(t)}$ is orthogonal to the span of $\Psi^{(t)} - \{\psi_i\}$, then the biorthogonal vector to $\psi_i^{(t)}$ is given by

$$\tilde{\psi}_i^{\perp(t)} = \psi_i^{\perp(t)} / \|\psi_i^{\perp(t)}\|^2 \tag{13}$$

Given the biorthogonal system, we can compute the current approximation to $\mathbf{x}$ as

$$\tilde{\mathbf{x}}^{(t)} = \sum_{i \in \mathcal{I}^{(t)}} \alpha_i^{(t)} \psi_i \quad \text{where} \quad \alpha_i^{(t)} = \langle \mathbf{x}, \tilde{\psi}_i^{\perp(t)} \rangle \tag{14}$$

The sign of the coefficients indicates whether or not an approximation lies in the convex cone of the atoms: if $\alpha_i < 0$ for some i then the approximation does not lie in the convex cone; the corresponding atom ψ_i is deleted from the representation and the biorthogonal system is updated.

The biorthogonal system and $\tilde{\mathbf{x}}^{(t)}$ can be updated as atoms are added to and subtracted from $\Psi^{(t)}$. Such adaptive biorthogonalization methods have recently been applied to MP [88, 7] and are standard in linear programming ([104], Chapter 8). We present pseudocode for adding an atom in Algorithm 2. and pseudocode for substracting an atom in Algorithm 3.

4.2 Analysis

By construction, GBP computes the minimum ℓ^1-norm representation of a given signal. To prove this we show that GBP converges to an exact representation in a finite number of steps and that the representation corresponds to a facet of the convex hull of the dictionary.

First, we prove that GBP converges to an exact representation. At each iteration of GBP there is a decrease in approximation error, as stated in the following theorem.

Algorithm 2 AddAtom

```
Input
  • The signal $\mathbf{x}$
  • The dictionary $\mathcal{D}$
  • The current representation $\mathcal{I}, \mathcal{A}$
  • The atom to add $k$, $\psi_k$
  • The current biorthogonal vectors $\tilde{\Psi}^\perp$
Output
  • The updated representation $\mathcal{I}, \mathcal{A}$
  • The updated biorthogonal vectors $\tilde{\Psi}^\perp$
Procedure
  1. Compute the new biorthogonal vector
     $\forall i \in \mathcal{I},\ \beta_i \leftarrow \langle \tilde{\psi}_i^\perp, \psi_k \rangle$
     $\psi_k^\perp \leftarrow \psi_k - \sum_{i \in \mathcal{I}} \beta_i \psi_i$
     $\tilde{\psi}_k^\perp \leftarrow \psi_k^\perp / \|\psi_k^\perp\|_2^2$
  2. Update the biorthogonal system
     $\forall i \in \mathcal{I},\ \tilde{\psi}_i^\perp \leftarrow \tilde{\psi}_i^\perp - \beta_i \tilde{\psi}_k^\perp$
     $\tilde{\Psi}^\perp \leftarrow \tilde{\Psi}^\perp \cup \{\tilde{\psi}_k^\perp\}$
  3. Update the representation
     $\alpha_k \leftarrow \langle \mathbf{x}, \tilde{\psi}_k^\perp \rangle$
     $\forall i \in \mathcal{I},\ \alpha_i \leftarrow \alpha_i - \beta_i \alpha_k$
     $\mathcal{I} \leftarrow \mathcal{I} \cup \{k\}$
     $\mathcal{A} \leftarrow \mathcal{A} \cup \{\alpha_k\}$
```

Theorem 1. *Given a signal $\mathbf{x} \in \mathbb{R}^d$ and a dictionary $\mathcal{D} = \{\psi_i\}_{i=1}^n$, where $n \geq 2d$, $\forall \psi_i \in \mathcal{D}$, $\psi_i \in \mathbb{R}^d$ and $\|\psi_i\|_2 = 1$, if $\psi_i \in \mathcal{D}$ then $-\psi_i \in \mathcal{D}$, and the atoms are in general position, if GBP is run with $\mathcal{D}$ and $\mathbf{x}$ as input and if $\tilde{\mathbf{x}}^{(t)} \neq 0$, then at iteration $t+1$ of GBP, $0 \leq \|\mathbf{x} - \tilde{\mathbf{x}}^{(t+1)}\|_2 < \|\mathbf{x} - \tilde{\mathbf{x}}^{(t)}\|_2$.*

Proof. At iteration t, let S be the hypersphere centered at $\mathbf{x}$ with radius $\|\mathbf{x} - \tilde{\mathbf{x}}^{(t)}\|_2$, let ψ_k be the next atom selected by GBP, and let T be the tangent plane to S at $\tilde{\mathbf{x}}^{(t)}$. T contains the origin (if it did not, then some scaling of $\tilde{\mathbf{x}}^{(t)}$ would be a better approximation to $\mathbf{x}$), and thus bisects the unit sphere. Because the atoms are in general position, $n \geq 2d$, and $\psi_i \in \mathcal{D}$ if $-\psi_i \in \mathcal{D}$, if $|\Psi^{(t)}| < d$, then there will be at least one atom in the same half-space of T as $\mathbf{x}$. (Note that if $|\Psi^{(t)}| = d$, we are done, as we would also have $\tilde{\mathbf{x}} = \mathbf{x}$.)

ψ_k lies in the same half-space of T as $\mathbf{x}$: by construction, there is no atom ψ_0 such that $\langle \psi_0 - \tilde{x}_H^{(t)}, \mathbf{n}^{(t)} \rangle > 0$, by general position, there is no atom ψ_0 such that $\langle \psi_0 - \tilde{x}_H^{(t)}, \mathbf{n}^{(t)} \rangle = 0$ and $\langle \psi_0 - \tilde{x}_H^{(t)}, \mathbf{v}^{(t)} \rangle > 0$, and, by the ordering of atoms by step 2(b) of GBP, GBP selects an atom in

```
Algorithm 3 SubtractAtom
Input
  • The signal $\mathbf{x}$
  • The current representation $\mathcal{I}, \mathcal{A}$
  • The index of the atom to subtract $k$
  • The current biorthogonal vectors $\tilde{\Psi}^{\perp}$
Output
  • The updated representation $\mathcal{I}, \mathcal{A}$,
  • The updated biorthogonal vectors $\tilde{\Psi}^{\perp}$
Procedure
  1. Delete the atom from the representation
     $\mathcal{I} \leftarrow \mathcal{I} - \{k\}$
     $\mathcal{A} \leftarrow \mathcal{A} - \{\alpha_k\}$
  2. Update the biorthogonal system
     $\tilde{\Psi}^{\perp} \leftarrow \tilde{\Psi}^{\perp} - \{\tilde{\psi}_k^{\perp}\}$
     $\forall i \in \mathcal{I},\ \gamma_i \leftarrow \langle \tilde{\psi}_k^{\perp}, \tilde{\psi}_i^{\perp} \rangle / \|\tilde{\psi}_k^{\perp}\|_2^2$
     $\forall i \in \mathcal{I},\ \tilde{\psi}_i^{\perp} \leftarrow \tilde{\psi}_i^{\perp} - \gamma_i \tilde{\psi}_k^{\perp}$
  3. Update the representation
     $\forall i \in \mathcal{I},\ \alpha_i \leftarrow \alpha_i - \alpha_k \gamma_i$
```

the same half-space of T as $\mathbf{x}$, if one exists.

ψ_k lies in the same half-space as $\mathbf{x}$, we can find a point $\tilde{\mathbf{x}} + \epsilon(\psi_k - \tilde{\mathbf{x}}^{(t)})$ that is interior to S and therefore closer to $\mathbf{x}$ than $\tilde{\mathbf{x}}^{(t)}$. Therefore $\|\mathbf{x} - \tilde{\mathbf{x}}^{(t+1)}\| < \|\mathbf{x} - \tilde{\mathbf{x}}^{(t)}\|$. □

Theorem 1 also implies that GBP does not cycle. GBP may select the same atom more than once, that is, GBP may select an atom, discard it, and select it again (this behaviour depends on the shape of the facets of $\mathbf{conv}(\mathcal{D})$), but GBP will never revisit the same state. Because there are a finite number of states and GBP improves at each iteration, GBP converges. By the same arguments as Theorem 1, at convergence the final supporting hyperplane contains a facet of $\mathbf{conv}(\mathcal{D})$ and thus GBP computes the minimum ℓ^1-norm representation.

The duality of GBP to the gravitational method [77] of linear programming, implies that the computationaly complexity of GBP is exponential in the worst-case [76]. Current results on the simplex algorithm suggest that GBP is likely to be polynomial in the average [15] and smoothed [96] cases.

4.3 Implementation Issues

We briefly describe two obstacles that any implementation of GBP may encounter, degeneracy and numerical instability, and our approach to handling them.

4.3.1 Degeneracy

Degeneracy occurs when the atoms of the dictionary are not in general position; the atoms are in general position if every k-face of $\mathbf{conv}(\mathcal{D})$ contains exactly $k+1$ atoms [107]. Degeneracy can occur if the dictionary is specially designed, for example, if the atoms are defined to be the vertices of a hypercube inscribed in the unit hypersphere. If GBP encounters degeneracy, the updates described in Section 4.1.3 will fail, resulting in an error. Although GBP does not currently include a mechanism to detect and handle degeneracy, incorporating such a feature is possible. A simple solution is to perturb the atoms of the dictionary sufficiently to place them in general position; see Section 5.1.

4.3.2 Numerical instability

Numerical instability can occur in the biorthogonalization stage of GBP. Let $\mathbf{\Psi}$ be a matrix corresponding to $\Psi^{(t)}$ for some t, where each row of $\mathbf{\Psi}$ is an atom in $\Psi^{(t)}$, and let $\tilde{\mathbf{\Psi}}^{\perp}$ denote the corresponding matrix of biorthogonal vectors. If at any iteration the matrix $\mathbf{\Psi}$ is ill-conditioned, the computation of the biorthogonal vectors we have described may be unstable (similar difficulties arise in Gram-Schmidt orthogonalization [89, 13]). One work around is to compute a full biorthogonalization at each iteration, or at least whenever instability is detected. However, a full biorthogonalization can be costly, as it is typically computed via the pseudoinverse [57]: since $\mathbf{\Psi}\left(\tilde{\mathbf{\Psi}}^{\perp}\right)^{T} = \mathbf{I}$, where $\mathbf{I}$ denotes the identity matrix, we can compute $\tilde{\mathbf{\Psi}}^{\perp}$ as $(\mathbf{\Psi}^{+})^{T}$, where '+' denotes the pseudoinverse.

We instead opt to compute the biorthogonalization using an iterative pseudoinverse technique [9]. This technique takes an initial estimate of the pseudoinverse and iteratively updates it, converging to the true pseudoinverse. If the initial estimate is sufficiently close to the true pseudoinverse, then the iterative pseudoinverse computation is substantially faster than the standard pseudoinverse. This approach is well suited to GBP, as the adaptive biorthogonalization already provides such an estimate.

The iterative pseudoinverse algorithm proceeds as follows. Given a matrix $\mathbf{\Psi}$ and an initial estimate of the pseudoinverse $\mathbf{\Psi}^{+(0)}$, the updates to $\mathbf{\Psi}^{+}$ are computed by

$$\mathbf{\Psi}^{+(t+1)} \leftarrow \mathbf{\Psi}^{+(t)} \left(2\mathbf{I} - \mathbf{\Psi}\mathbf{\Psi}^{+(t)}\right)$$

(Note that here t denotes the iteration of the pseudoinverse algorithm, not the iteration of GBP.) For a detailed analysis of this algorithm, see [95]. While a classic technique, this algorithm is the subject of ongoing research [82, 85].

Our implementation of GBP tests if $\mathbf{\Psi}\left(\tilde{\mathbf{\Psi}}^{\perp}\right)^{T} = \mathbf{I}$ within a specified level of tolerance after each adaptive biorthogonalization. If the test fails, the iterative pseudoinverse algorithm is applied.

5 Results

We examine the performance of GBP. We compared the running time of GBP to that of standard linear programming algorithms on three data sets, random data, speech data, and seismic data, described below. We also provide an example of GBP's performance on a single signal and contrast it with that of Matching Pursuit.

In each experiment, we measured the running times of GBP and standard linear programming algorithms on the signal representation problems described below. The algorithms we compared were GBP, two variants of the simplex method, and an interior point method.

The implementation of GBP used was our own, written entirely in Matlab. The linear programming solvers used were those included in the Matlab Optimization Toolbox 3.0 [4], and a freely available Matlab implementation [75] of the revised simplex method [27]. The Optimization Toolbox version of the simplex method is the classical simplex method [28], with the initial basis determined as in [11]. The Optimization Toolbox version of the interior point method is essentially LIPSOL [106], a freely available interior point solver that implements Mehrotra's predictor-corrector method [73, 70].

For each problem, all algorithms were run and timed. All algorithms were run under Matlab 7 on a 1.5GHz Pentium M processor running Windows XP, with 1.25GB memory. On all problems all algorithms returned identical representations (up to the specified error tolerance).

5.1 Running times: Random data

The random data set consisted of 3000 randomly generated signal representation problems, varying both the dimension of the signal space and the overcompleteness of the dictionary. Each problem consisted of a randomly generated signal and a randomly generated dictionary. The dimension d of the problems varied over the set $\{8, 16, 32, 64, 128, 256\}$. The overcompleteness k of the dictionaries varied over the set $\{2, 4, 8, 16, 32\}$. In each problem, the signal $\mathbf{x}$ was randomly generated to be uniformly distributed on the unit hypersphere in $\mathbb{R}^d$. The dictionary for each problem had $2kd$ atoms; the first kd of these atoms were generated in the same fashion as the signal, the second kd atoms were the negatives of the first kd atoms. Additionally, the dictionary of each problem was perturbed: To each atom was added Gaussian noise with variance 0.000001, after which the atom was normalized to lie on the unit hypersphere; this perurbation ensures that the linear programming algorithms can compute the requisite matrix inverses; for structured dictionaries this perturbation also ensures that the atoms are in general position. For each d-k pair, 100 problems were generated.

Figure 4 shows running times of the three algorithms as a function of overcompleteness for each dimension; the curve plotted shows the mean running time of each algorithm over the 100 problems of the specified dimension and overcompleteness, with error bars showing the corresponding minimum and maximum running times. (We do not show the results of the revised simplex method here, as it was outperformed by the Matlab's simplex algorithm.)

5.2 Running times: Speech data

The speech data set consisted of 100 signal representation problems. Each problem consisted of a signal randomly drawn from the TIMIT database [53] and an overcomplete multiscale Gabor dictionary.

Each signal comprised 256 samples ($d = 256$) and was randomly selected from the 'train' subset

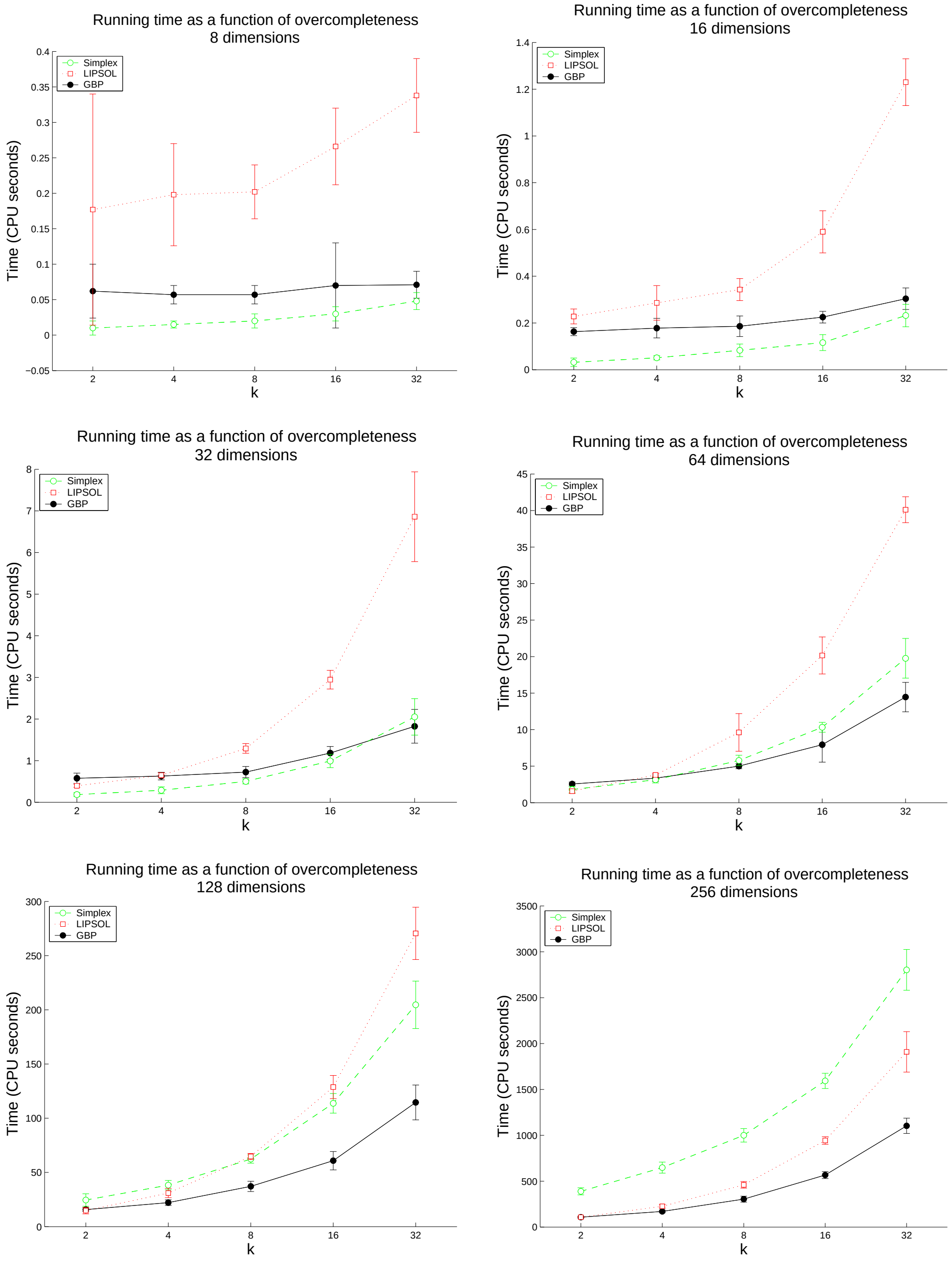


Figure 4: Running times for GBP, LIPSOL, and the simplex method (Matlab) on the random data set, plotted as a function of overcompleteness for each dimension. Note that GBP's performance improves relative to the other methods as the dimensionality of the problems increase.

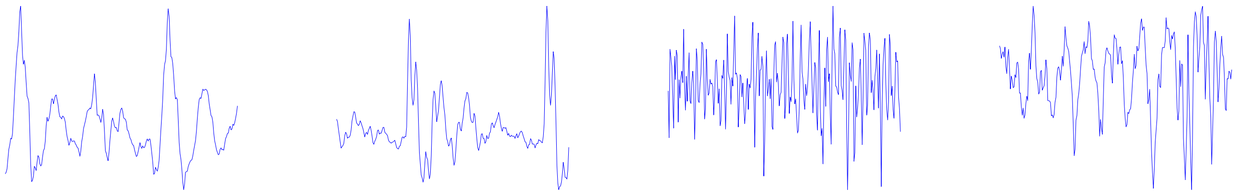


Figure 5: Four signals drawn from the speech data set. Each signal consists of 256 samples.

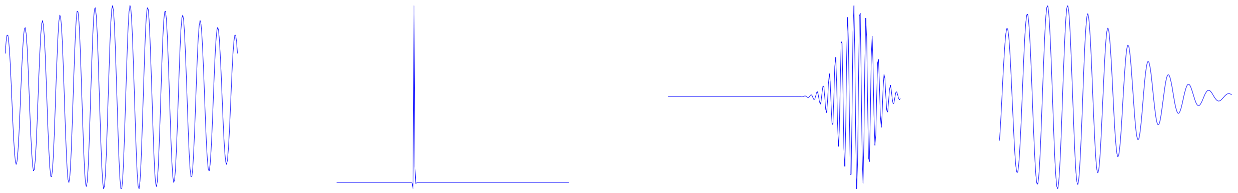


Figure 6: Four atoms drawn from the multiscale Gabor cosine dictionary.

of the TIMIT database. The signals were mean centered and normalized. Some samples are shown in Figure 5.

The dictionary used was a 9×overcomplete multiscale Gabor dictionary (4608 atoms). The dictionary consisted of several fixed scale critically sampled cosine Gabor bases. Each atom was defined by the parameters t_0 and f as $G(t; t_0, f) = (1/(2\pi\sigma)) \exp^{-(t-t_0)^2/\sigma^2} \cos(2\pi f(t - t_0))$, where $t_0 \in \{0 : \Delta t : 1\}$ and $f \in \{0 : \Delta f : d/2\}$, with $\Delta t = 2^j/d$, $\sigma = \sqrt{\pi/2}/\Delta t$, and $\Delta f = \sigma/\sqrt{2\pi}$; the scale parameter j varied over $\{0, 1, \ldots, 8\}$. See [52, 45] for details and other sampling schemes. Once the atoms were defined, they were perturbed as in the random data case. Some samples from the final dictionary are shown in Figure 6.

We show the running times of GBP, LIPSOL, and the revised simplex method on the sound data set in Table 1. (The revised simplex method outperformed Matlab's simplex method.) We show the mean, minimum, and maximum runninng times for each algorithm on the 100 signals.

5.3 Running times: Seismic data

The seismic data consists of 100 signal representation problem. Each problem consists of a 256 sample signal of seismic recordings from the North Sea, 4 times downsampled from the original data [94]; some samples are shown in Figure 7. The dictionary used was the same as used in the

Algorithm	Min	Mean	Max
GBP	40.41	48.72	56.35
LIPSOL	58.62	75.97	155.56
Revised Simplex	441.66	1297.65	2700.51

Table 1: Running times of GBP, LIPSOL, and the revised simplex method on the sound data set, in CPU seconds.

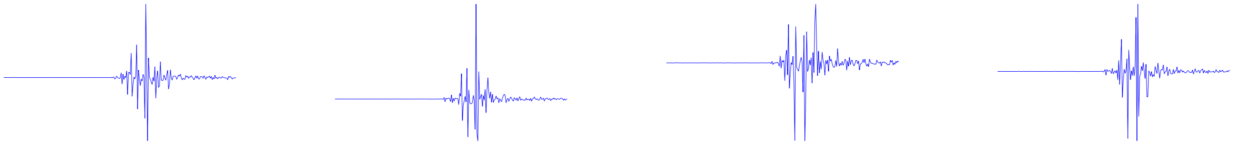


Figure 7: Four signals drawn from the seismic data set. Each signal consists of 256 samples.

speech experiment above. We show the running times of GBP, LIPSOL, and the revised simplex method on the seismic data set in Table 2.

5.4 Example: Speech signal

Figure 8 provides an example comparing GBP to MP and OMP on a 1024-dimensional signal (Figure 8, top left), selected from the TIMIT speech database [53], using a multiscale Gabor dictionary ($n = 22528$), similar to the one used for the sound data. (Note that the other BP methods were unable to compute representations on problems of this size in our environment.) Examining the

Algorithm	Min	Mean	Max
GBP	42.29	48.83	55.05
LIPSOL	60.52	70.36	112.90
Revised Simplex	2233.45	2489.05	2831.59

Table 2: Running times of GBP, LIPSOL, and the revised simplex method on the seismic data set, in CPU seconds.

approximation error of each algorithm as a function of iteration (Figure 8, top right), we observe that while the approximation error of GBP decreases somewhat more slowly than that of MP (note also that each iteration of GBP is more costly), the error of GBP does appear to decrease approximately exponentially. Also note that the representation computed by GBP is sparser than that of MP, though less sparse than that of OMP, as indicated by the sorted-amplitudes-curves and the ℓ^1-norm of the representations. The sorted-amplitudes-curves [66, 62] (Figure 8, bottom left) are plots of the logarithm of the final coefficients, sorted in descending order; the rates of decrease indicate the relative sparsity of the representations. The ℓ^1-norm of the representation coefficients are 0.3274, 0.4569, and 0.4156 for GBP, MP, and OMP respectively. (Note that the results for GBP would be the same as those for standard linear programming methods for BP.) The feature of GBP to note here is its 'greediness': the coefficients in the order of atom selection track the sorted-amplitudes-curve, that is, GBP tends to select significant atoms early on (Figure 8, bottom right). This demonstrates that it is possible to compute Basis Pursuit signal representations and to be greedy at the same time.

6 Discussion

Our results show that GBP provides a fast alternative to standard linear programming methods for sparse signal representation problems, particularly when the dimension of the signal space is high and the dictionary is very overcomplete. While there are a variety of factors which may contribute to the results, there are several algorithmic reasons why we expect GBP to perform well relatively.

The efficient solution of linear programming problems depends in a complicated way on the problem, the method of solution and its implementation, and the available resources; see Bixby [12]. Thus the relative success of GBP compared to the linear programming methods implemented in the Matlab Optimization Toolbox is partially a function of the specific methods used and their implementation. There are many available linear programming packages [48], some specific to sparse representation include Atomizer [1], ℓ_1-MAGIC [2], and SparseLab [3]. An exhaustive comparison of GBP against all of these methods is out of the scope of the present paper. However, while

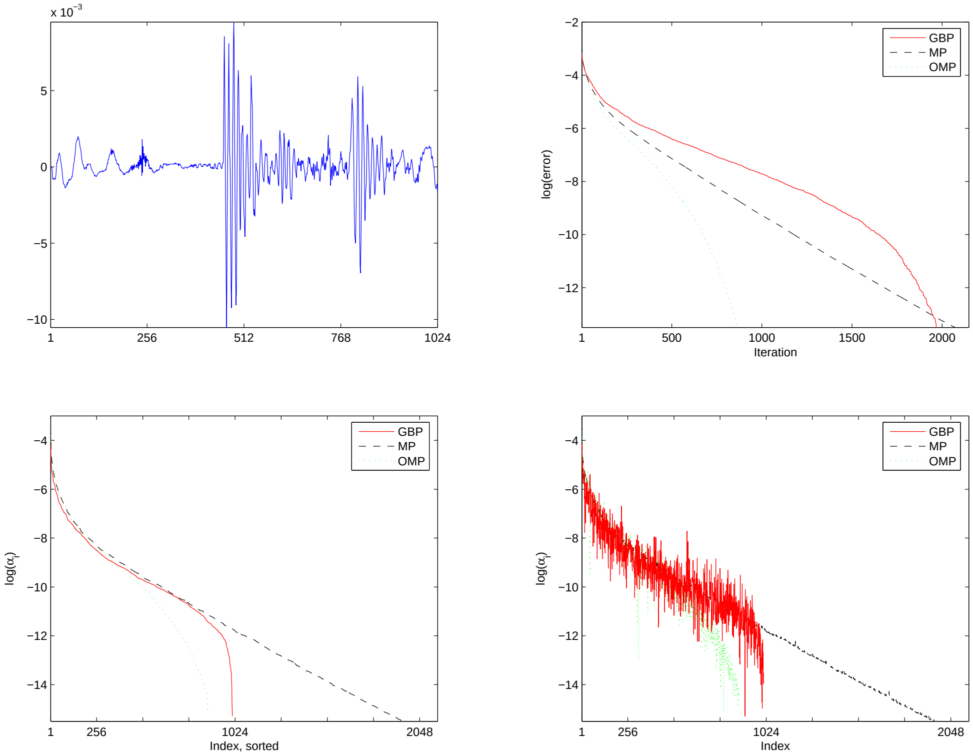


Figure 8: An example comparing GBP to MP on a speech signal (a). (b) The log of the error (ℓ^2-norm of the residual) as a function of iteration. (c) The sorted-amplitudes-curves; observe that GBP produces a sparser representation than MP. (d) The (final) coefficient values, in order of atom selection. (Note that the coefficient values change in GBP at each iteration.) See text for discussion.

the linear programming methods against which GBP was compared may not represent the current state-of-the-art, it is worth noting that GBP itself has the potential for significant speed increases through more efficient implementation.

Algorithmically, GBP has several advantages over standard linear program solvers. First, most linear program solvers assume, for historical reasons, that the constraint matrix is sparse, and they therefore rely on techniques that exploit this sparsity, whether or not sparsity is actually present [22]. The signal representation problems considered here are not particularly sparse in this sense: the random dictionary matrices used in the random data set are certainly not sparse, while the Gabor dictionary matrices used with the sound and seismic data sets are somewhat sparse, however the matrices are not sufficiently structured for certain fast algorithms to be applicable [20, 21]. GBP does not exploit this sparsity, and therefore does not suffer when it is not present. Second, GBP is efficient in the search for the next atom to select, because this search is based on a geometric criterion that involves 2 projections per possible atom. Simplex methods can be inefficient at this task as the search can involve evaluating more than 2, even d, projections per possible atom; see [104, 101]. Third, the updates in GBP are seldom of a full basis, further reducing computation. Finally, the complexity of the simplex method depends on the closeness of the initial solution to the optimal solution, which in turn depends on the phase I algorithm by which the initial basis is selected. GBP does not depend on an intial solution; in fact, GBP can be interpreted as a combined phase I/ phase II linear programming algorithm.

One area which we have not explored that merits further investigation is the dependence of the performance of GBP (and other sparse representation algorithms) on the structure of the dictionary. For example, a dictionary optimized for use with MP [30] or OMP [44] may well have very different properties from one optimized for BP. The design of dictionaries has only recently received attention in the signal processing community [30, 44, 5] (for work in neural computation, see [80, 69]); our work suggests that the geometric properties of dictionaries play a crucial role in both the efficiency of representation algorithms and the quality of the resulting representations. Indeed, geometric considerations have already led to a better theoretical understanding of sparse signal representation [39, 38].

As noted, part of the motivation for the development of BP is the observation that MP and OMP can fail to find sparse, in the ℓ^0-norm sense, signal representations [21], with much theoretical work showing under exactly what conditions BP finds sparse representations, i.e., when the minimal ℓ^1-norm solution is equivalent to the minimal ℓ^0-norm solution [37, 36, 51]. These findings have made BP useful for areas beyond signal representation, including compressed sensing [35] and error correcting codes [16], thus GBP may prove useful in these domains.

7 Conclusions

We have described GBP, a new algorithm for Basis Pursuit, and demonstrated that it is faster than standard linear programming methods on some problems, particularly in high-dimensional signal spaces using very overcomplete dictionaries. A Matlab implementation of GBP is currently available online at: `http://www.cs.yale.edu/~huggins/gbp.html`

Computational geometry has traditionally been the preserve of computer science, particularly computer graphics and theoretical computer science; its use here in the development of GBP highlights the relevance of computational geometry to signal processing. GBP also illustrates the interplay between signal processing and linear programming. That an efficient linear programming algorithm falls naturally out of sparse signal representation is surprising, and suggests that researchers in signal processing should not view linear programming, or optimization in general, as a black box: on one hand signal processing naturally defines a set of problems that can serve to drive research in linear programming, on the other hand, given the historical parallels, optimization research deserves deeper examination by the signal processing community.

Acknowledgements

We thank Mauro Maggioni for many helpful discussions, Ohad Ben-Shahar, Pavel Dimitrov, and Gang Li for their advice, Karl Skretting for providing the seismic data, and Dan Spielman for his insight into linear programming. We also thank the three anonymous reviewers for their suggestions which have considerably improved the article. Research supported by DARPA and AFOSR.

References

[1] `http://www-stat.stanford.edu/~atomizer`.

[2] `http://www.acm.caltech.edu/l1magic`.

[3] `http://sparselab.stanford.edu`.

[4] *Optimization Toolbox User's Guide.* The MathWorks, 2003.

[5] M. Aharon, M. Elad, A.M. Bruckstein, and Y. Katz. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. On Signal Processing*, In press.

[6] O.K. Al-Shaykh, E. Miloslavsky, T. Nomura, R. Neff, and A. Zakhor. Video compression using matching pursuits. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(1), 1999.

[7] M. Andrle, L. Rebollo-Neira, and E. Sagianos. Backward-optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 2004.

[8] J.B. Bednar, R. Yarlagadda, and T. Watt. L_1 deconvolution and its application to seismic signal processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(6), 1986.

[9] A. Ben-Israel. An iterative method for computing the generalized inverse of an arbitrary matrix. *Mathematics of Computation*, 19:452–455, 1965.

[10] F. Bergeaud and S. Mallat. Matching pursuit of images. In *Proceedings of the International Conference on Image Processing*, 1995.

[11] R.E. Bixby. Implementing the simplex method: The initial basis. *ORSA Journal on Computing*, 4(3), 1992.

[12] R.E. Bixby. Solving real-world linear programs: A decade and more of progress. *Operations Research*, 50(1):3–15, 2002.

[13] A Björck. Numerics of Gram-Schmidt orthogonalization. *Linear Algebra and its Applications*, 197-198:297–316, 1994.

[14] P. Bofill and M. Zibulevsky. Underdetermined blind source separation using sparse representations. *Signal Processing*, 81:2353–2362, 2001.

[15] K.H. Borgwardt. *The Simplex Method - A Probabilistic Analysis*. Springer-Verlag, 1987.

[16] E. Candes, M. Rudelson, R. Vershynin, and T. Tao. Error correction via linear programming. In *FOCS*, 2005.

[17] D.R. Chand and S.S. Kapur. An algorithm for convex polytopes. *Journal of the Association for Computing Machinery*, 17(1):78–86, 1970.

[18] S.Y. Chang and K.G. Murty. The steepest descent gravitational method for linear programming. *Discrete Applied Mathematics*, 25:211–239, 1989.

[19] S. Chen and D. Donoho. Basis pursuit. In *Twenty-Eighth Asilomar Conference on Signals, Systems & Computers*, 1994.

[20] S.S. Chen. *Basis Pursuit*. PhD thesis, Stanford University, Department of Statistics, 1995.

[21] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[22] V. Chvátal. *Linear Programming*. Freeman, 1983.

[23] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. McGraw-Hill, 1990.

[24] S.F. Cotter, J. Adler, B.D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. *IEE Proc.-Vis. Signal Processing*, 146(5), 1999.

[25] S.F. Cotter, K. Kreutz-Delgado, and B.D. Rao. Backward sequential elimination for sparse vector subset selection. *Signal Processing*, 81:1849–1864, 2001.

[26] G.B. Dantzig. Converting a converging algorithm into a polynomial bounded algorithm. Technical Report SOL 91-5, Systems Optimization Laboratory, Stanford University, 1991.

[27] G.B. Dantzig and W. Orchard-Hays. The product form for the inverse in the simplex method. *Mathematical Tables and Other Aids to Computation*, 8:64–67, 1954.

[28] G.B. Dantzig, A. Orden, and P. Wolfe. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific Journal of Mathematics*, 5:183–195, 1955.

[29] I. Daubechies. Time-frequency localization operators: A geometric phase space approach. *IEEE Transactions on Information Theory*, 34(4), 1988.

[30] G. Davis. *Adaptive Nonlinear Approximations*. PhD thesis, New York University, Department of Mathematics, 1994.

[31] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximation. *Constructive Approximation*, 13:57–98, 1997.

[32] G. Davis, S. Mallat, and Z. Zhang. Adaptive time-frequency decompositions with matching pursuit. *Optical Engineering*, 33(7):2183–2191, 1994.

[33] R.A. DeVore and V.N. Temlyakov. Some remarks on greedy algorithms. *Advances in Computational Mathematics*, 5:173–187, 1996.

[34] D.L. Donoho. Sparse components of images and optimal atomic decompositions. *Constructive Approximation*, 17(3):353–382, 2001.

[35] D.L. Donoho. Compressed sensing. Technical report, Stanford University, Department of Statistics, September 2004.

[36] D.L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *PNAS*, 100(5):2197–2202, 2003.

[37] D.L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7), 2001.

[38] D.L. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences*, 102(27):9452–9457, 2005.

[39] D.L. Donoho and J. Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences*, 102(27):9446–9451, 2005.

[40] B. Drachman. Two methods to deconvolve: L_1-method using simplex algorithm and L_2-method using least squares and a parameter. *IEEE Transactions on Antennas and Propagation*, 32(3), 1984.

[41] M.A. Efroymson. Multiple regression analysis. In A. Ralston and H.S. Wilf, editors, *Mathematical Methods for Digital Computers*, pages 191–203. Wiley, 1960.

[42] M. Elad. Why simple shrinkage is still relevant for redundant representations? *IEEE Transactions on Information Theory*. To appear.

[43] L.O. Endelt and A. la Cour-Harbo. Comparison of methods for sparse representation of music signals. In *Proccedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[44] K. Engan, S.O. Aase, and J.H. Husøy. Multi-frame compression: Theory and deisgn. *Signal Processing*, 80(10):2121–2140, 2001.

[45] H.G. Feichtinger and T. Strohmer, editors. *Gabor Analysis and Algorithms, Theory and Applications.* Birkhäuser, 1998.

[46] H.G. Feichtinger, A. Türk, and T. Strohmer. Hierarchical parallel matching pursuit. In *Proc. SPIE: Image Reconstruction and Restoration*, pages 222–232, 1994.

[47] S.E. Ferrando, E.J. Doolittle, A.J. Bernal, and L.J. Bernal. Probabilistic matching pursuit with Gabor dictionaries. *Signal Processing*, 80:2099–2120, 2000.

[48] R. Fourer. Software survey: Linear programming. *OR/MS Today*, June 2005.

[49] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376), 1981.

[50] J.H. Friedman and J.W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, C-23:881–890, 1974.

[51] J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Trans. on Information Theory*, 50(6), 2004.

[52] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[53] J.S. Garofolo, L.F. Lamel, W.M. Fisher, J.G. Fiscus, D.S. Pallett, and N.L. Dahlgren. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. Technical Report NISTIR-4930, US Dept. of Commerce, National Institute of Standards and Technology, 1993.

[54] M. Gharavi-Alkhansari and T.S. Huang. A fast orthogonal matching pursuit algorithm. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1389–1392, 1998.

[55] A.C. Gilbert and J.A. Tropp. Applications of sparse approximations in communications. In *Proceedings of IEEE International Symposium on Information Theory*, 2005.

[56] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Trans. on Signal Processing*, 45(3), 1997.

[57] T.N.E. Greville. The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations. *SIAM Review*, 1(1), 1959.

[58] R. Gribonval and P. Vandergheynst. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Trans. on Information Theory*, 52(1), 2006.

[59] G. Harikumar, C. Couvreur, and Y. Bresler. Fast optimal and suboptimal algorithms for sparse solutions to linear inverse problems. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1877–1881, 1998.

[60] A.J. Hoffman. On greedy algorithms that succeed. In I. Anderson, editor, *Surveys in Combinatorics 1985*, pages 97–112, 1985.

[61] P.J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[62] X. Huo. *Sparse Image Representation via Combined Transforms*. PhD thesis, Stanford University, Department of Statistics, 1999.

[63] S. Jaggi, W.C. Karl, S. Mallat, and A.S. Willsky. High resolution pursuit for feature extraction. *Applied and Computational Harmonic Analysis*, 5:428–449, 1998.

[64] R.A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2:18–21, 1973.

[65] J. Karvanen and A. Cichocki. Measuring sparseness of noisy signals. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 125–130, 2003.

[66] K. Kreutz-Delgado and B.D. Rao. Measures and algorithms for best basis selection. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998.

[67] S.A. Leichner, G.B. Dantzig, and J.W. Davis. A strictly improving phase I algorithm using least-squares subproblems. Technical Report SOL 92-1, Systems Optimization Laboratory, Stanford University, 1992.

[68] M.S. Lewicki. Efficient coding of natural sounds. *Nature Neuroscience*, 5(4):356–363, 2002.

[69] M.S. Lewicki and T.J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 2000.

[70] I.J. Lustig, R.E. Marsten, and D.F. Shanno. On implementing Mehrotra's predictor-corrector interior point method for linear programming. *SIAM J. Optimization*, 2(3):435–449, 1992.

[71] D. Malioutov, M. Cetin, and A.S. Willsky. A sparse signal reconstruction perspective on source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8), 2005.

[72] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12), 1993.

[73] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[74] A.J. Miller. *Subset Selection in Regression, Second Edition*. CRC Press, 2002.

[75] S.S. Morgan. A comparison of simplex method algorithms. Master's thesis, University of Florida, 1997.

[76] T.L. Morin, N. Prabhu, and Z. Zhang. Complexity of the gravitational method for linear programming. *Journal of Optimization Theory and Applications*, 108(3):633–658, 2001.

[77] K.G. Murty. The gravitational method for linear programming. *Opsearch*, 23:206–214, 1986.

[78] B.K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[79] M.S. O'Brien, A.N. Sinclair, and S.M. Kramer. Recovery of a sparse spike time series by L_1 norm deconvolution. *IEEE Transactions on Signal Processing*, 42(12), 1994.

[80] B.A. Olshausen and D.J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[81] P.-Q. Pan. A basis-deficiency-allowing variation of the simplex method for linear programming. *Computers and Mathematics with Applications*, 36(3), 1998.

[82] V. Pan and R. Schreiber. An improved Newton iteration for the generalized inverse of a matrix, with applications. *SIAM J. Sci. Stat. Comput.*, 12(5):1109–1130, 1991.

[83] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Twenty-Seventh Asilomar Conference on Signals, Systems & Computers*, 1993.

[84] P.J. Phillips. Matching pursuit filters applied to face identification. *IEEE Transactions on Image Processing*, 7(8), 1998.

[85] W.H. Pierce. A self-correcting matrix iteration for the Moore-Penrose generalized inverse. *Linear Algebra and Its Applications*, 244:357–363, 1996.

[86] S. Qian and D. Chen. Signal representation using adaptive normalized Gaussian functions. *Signal Processing*, 36:1–11, 1994.

[87] B.D. Rao. Signal processing with the sparseness constraint. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1998.

[88] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. *IEEE Signal Processing Letters*, 9(4), 2002.

[89] J.R. Rice. Experiments on Gram-Schmidt orthogonalization. *Math. Comp.*, 20:325–328, 1966.

[90] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proc.18th ACM Symposium on the Theory of Computation*, pages 404–413, 1986.

[91] R. Seidel. Convex hull computations. In J.E. . E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry, Second Edition*. CRC Press, 2004.

[92] A.P. Sethi and G.L. Thompson. The pivot and probe algorithm for solving a linear program. *Mathematical Programming*, 29:219–233, 1984.

[93] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. Shiftable multi-scale transforms. *IEEE Transactions on Information Theory*, 38(2), 1992.

[94] K. Skretting. Pre-stack/post-stack seismic data from the North Sea. `http://www.ux.his.no/ karlsk/sdata/`.

[95] T. Söderström and G.W. Stewart. On the numerical properties of an iterative method for computing the Moore-Penrose generalized inverse. *SIAM J. Numer. Anal.*, 11(1), 1974.

[96] D.A. Spielman and S. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. In *STOC'01*, pages 296–305, 2001.

[97] J.L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component separation. *Advances in Imaging and Electron Physics*, 2004.

[98] J.J. Stone. The cross-section method. Technical Report P-1490, The RAND Corporation, 1958.

[99] G. Swart. Finding the convex hull facet by facet. *Journal of Algorithms*, 6:17–48, 1985.

[100] V.N. Temlyakov. Greedy algorithms and m-term approximation with regard to redundant dictionaries. *Journal of Approximation Theory*, 98(1):117–145, 1999.

[101] T. Terlaky and S. Zhang. A survey on pivot rules for linear programming. Technical Report 91-99, Delft University of Technology, Faculty of Technical Mathematics and Informatics, 1991.

[102] J.A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10), 2004.

[103] J.A. Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 52(3), 2006.

[104] R.J. Vanderbei. *Linear Programming: Foundations and Extensions, Second Edition*. Kluwer Academic Publishers, 2001.

[105] P. Vandergheynst and P. Frossard. Image coding using redundant dictionaries. In M. Barni, editor, *Document and Image Compression*. CRC Press, 2006.

[106] Y. Zhang. User's guide to LIPSOL Linear-programming Interior Point SOLvers v0.4. *Optimization Methods and Software*, 11-12, 1999.

[107] G.M. Ziegler. *Lectures on Polytopes*. Springer, 1994.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."
"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Data fusion and multi-cue data matching by diffusion maps

Stéphane Lafon, Yosi Keller, Andreas Glaser and Ronald R. Coifman

Abstract

Data fusion and multi-cue data matching are fundamental tasks in high dimensional data analysis. In this paper, we apply the recently introduced diffusion framework to address these tasks. Our contribution is three-fold. First, we present the Laplace-Beltrami approach to computing density invariant embeddings which are essential for integrating different sources of data. Second, we describe a refinement of the Nyström extension algorithm called “geometric harmonics”. We also explain how to use this tool for data assimilation. Finally, we introduce a multi-cue data matching scheme based on nonlinear spectral graphs alignment.

The effectiveness of the proposed scheme is validated by applying it to the lip reading and image sequence alignment.

Program of Applied Mathematics, Department of Mathematics, Yale University

New Haven, CT 06520.

Email: {stephane.lafon, yosi.keller, andreas.glaser, coifman-ronald}@yale.edu

Index Terms

data fusion, data matching, spectral graph theory, diffusion processes, machine learning, graph algorithms, information visualization, data mining

I. Introduction

The processing of massive high-dimensional data sets is a contemporary challenge. Suppose that a source s produces high-dimensional data $\{x_1, ..., x_n\}$ that we wish to analyze. For instance, each data point could be the frames of a movie produced by a digital camera, or the pixels of a hyperspectral image. When dealing with this type of data, the high-dimensionality is an obstacle for any efficient processing of the data. Indeed, many classical data processing algorithms have a computational complexity that grows exponentially with the dimension (this is the so-called "curse of dimensionality"). On the other hand, the source s may have a limited number of degrees of freedom. In this case, the high dimensional representation of the data is an unfortunate (but often necessary) artifact of the choice of sensors or the acquisition device. This means that the data have a low intrinsic dimensionality, or equivalently, that many of the variables that describe each data points are highly correlated, at least locally. Therefore it is possible to obtain low-dimensional representations of the samples. Note that since the variables are correlated only locally, classical global dimension reduction methods like Principal Component Analysis and Multidimensional Scaling do not provide, in general, an efficient dimension reduction.

First introduced in the context of manifold learning, eigenmaps techniques [1], [2], [3], [4] are becoming increasingly popular as they overcome this problem. Indeed, they perform a nonlinear reduction of the dimension by providing a parametrization of the data set that preserves neighborhoods. However, the new representation that one obtains is highly sensitive to the way the data points were originally sampled. More precisely, if the data are assumed to approximately lie on a manifold, then the eigenmap representation depends on the density of the points on this manifold [5]. This issue is of critical importance in applications as one often needs to *merge data* that were produced by the same source but acquired with different devices or sensors, at various sampling rates and possibly on different occasions. In that case, it is necessary to have a canonical representation of the data that retains the intrinsic constraints of the samples (e.g. manifold geometry) regardless of the particular distribution of the datasets sampled by different devices.

Another important issue is that of *data matching*. This question arises when one needs to establish a correspondence between two data sets resulting from the same fundamental source. For instance, consider the problem of matching pixels of a stereo image pair. One can form a graph for each image, where pixels constitute the nodes, and where edges are weighted according to the local features in the image. The problem now boils down to matching nodes between two graphs. Note that this situation is an instance of multi-sensor integration problem, in which one needs to find the correspondence between data captured by different sensors. In some applications, like fraud detection, synchronizing data sets is used for detecting discrepancies rather than similarities between data sets.

The out-of-sample extension problem is another aspect of the data fusion problem. The idea is to extend a function known on a training set to a new point using both the target function and the geometry of training domain. The new point and the corresponding value of the function can then be assimilated to the training set. This is an essential component in any scheme that agglomerates knowledge over an initial data set and then applies the inferred structure to new data. Recently, Belkin *et al* have developed a solution to this problem via the concept of manifold regularization [6]. Earlier, several authors used the Nyström extension procedure in the Machine Learning context [7], [8] in order to extend eigenmap coordinates. In both cases, the question of the scale of the extension kernel remains unanswered. In other words, given an empirical function on a data set, to what distance to the training set can this function be extended ? In particular, given the spectral embedding of the data set, which kernel should be used to extend it?

By relating the frequency content of the target function on the training set to the extrinsic Fourier analysis, Coifman *et al* provide an answer to this question [9]. They developed the idea of “geometric harmonics” based on the Nyström extension at different scales, providing a multiscale extension scheme for empirical functions. We apply this concept to the extension of spectral embeddings and show that the extension has to be conducted using a specially designed kernel which differs from data embedding kernel.

In this article, we show that the questions discussed above can be efficiently addressed by the general diffusion framework introduced in [10], [11]. The main idea is that, just like for eigenmaps methods, eigenvectors of Markov matrices can be used to embed any graph into a Euclidean space and achieve dimension reduction. Building on these ideas, the contribution of this paper is three-fold: first, we show that by carefully normalizing the Markov matrix, the embedding can be made invariant to the density of the sampled data points, thus solving the problem of data fusion encountered with other eigenmaps methods. Then, we address the problem of out-of-sample extension, and we explain how to extend empirical functions to new samples using the geometric harmonics. Last, we take advantage of the density-invariant representation of data sets provided by the diffusion coordinates to derive a simple data matching algorithm based on geometrical embeddings alignment.

The proposed scheme is experimentally verified by applying it to visual data analysis. First, we address the problem of automatic lip reading by embedding the lips images using the Laplace-Beltrami and deriving an automatic lip reading scheme where new data is assimilated using geometric harmonics. Second, we demonstrate the multi-cue data matching aspect of our work by matching image sequences corresponding to similar head motions.

This paper is organized as follows: we start by recalling the diffusion framework, and the notion of diffusion maps in Section II. We then explain in Section II-B how to normalize the diffusion kernel in order to separate the geometry (constraints) of the data from the distribution of the points. We describe the out-of-sample extension procedure via the geometric harmonics in Section II-D and present a nonlinear algorithms for matching two data sets. Last, we illustrate these ideas by applying it to lip reading and sequence alignment in Section III.

II. The diffusion framework

A. *Diffusion maps and diffusion distances*

Let $\Omega = \{x_1, ..., x_n\}$ be n data points. In this section, we recall the diffusion framework as described in [5], [12]. The main point of this set of techniques is to introduce a useful metric on data sets based on the connectivity of points within the graph of the data, and also to provide coordinates on the data set that reorganize the points according to this metric.

The first step in our construction is to view these points as being the nodes of a symmetric graph in which two nodes are connected by an edge if they are very similar. The very notion of similarity between two data points is completely application-driven. In many situations, each data point is a collection of numerical measurements and can be thought of as a point in a Euclidean feature space. In this case, similarity is measured in terms of closeness in this space, and it is custom to weight the edge between x_i and x_j by $\exp(\|x_i - x_j\|^2/\varepsilon)$, where $\varepsilon > 0$ is a scale parameter. More generally, we allow ourselves to consider arbitrary weight functions $w(\cdot, \cdot)$ that verify the following two conditions, for all x and y in Ω:

- it is symmetric: $w(x, y) = w(x, y)$,
- it is pointwise non-negative: $w(x, y) \geq 0$.

The weight function or kernel describes the first-order interaction between the data points as it defines the nearest neighbor structures in the graph. The analysis of the data provided by the diffusion techniques depends heavily on the choice of the weight function.

Following a classical construction in spectral graph theory [13], namely the normalized graph Laplacian, we now create a random walk on the data set Ω by forming the following kernel:

$$p_1(x, y) = \frac{w(x, y)}{d(x)} \,,$$

where $d(x) = \sum_{z \in \Omega} w(x, z)$ is the degree of node x.

Since we have that $p_1(x, y) \geq 0$ and $\sum_{y \in \Omega} p_1(x, y) = 1$, the quantity $p_1(x, y)$ can be interpreted as the probability of a random walker to jump from x to y in a single time step. If P is the $n \times n$ matrix of transition of this Markov chain, then taking powers of this matrix amounts to running the chain forward in time. Let $p_t(\cdot, \cdot)$ be the kernel corresponding to the t^{th} power of the matrix P. In other words, $p_t(\cdot, \cdot)$ describes the probabilities of transition in t time steps.

The asymptotic behavior of this random walk has been used to find clusters in the data set [13], [14] where the first non-constant eigenfunction is used as a classification function into two clusters. More recently, using the other eigenfunctions was considered [15]. For $t = +\infty$, this Markov chain is governed by a unique stationary distribution ϕ_0, which means that

$$\lim_{t \to +\infty} p_t(x, y) = \phi_0(y) \,.$$

The vector ϕ_0 is the top left eigenvector of P, *i.e.*, $\phi_0^T P = \phi_0^T$, and it can be checked that $\phi_0(y)$ is given by

$$\phi_0(y) = \frac{d(y)}{\sum_{z\in\Omega} d(z)} \,.$$

It can be shown [12] that the pre-asymptotic regime is governed according to the following eigendecomposition

$$p_t(x,y) = \sum_{l\geq 0} \lambda_l^t \psi_l(x)\phi_l(y) \,, \tag{1}$$

where $\{\lambda_l\}$ is the sequence of eigenvalues of P (with $|\lambda_0| \geq |\lambda_1| \geq ...$) and $\{\phi_l\}$ and $\{\psi_l\}$ are the corresponding left and right eigenvectors (see the appendix for a proof). Furthermore, because of the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in the previous sum. Let $m(t)$ be this number.

Unifying ideas from Markov chains and potential theory, the diffusion distance between two points x and z was introduced in [12], [5] as

$$D_t^2(x,z) = \sum_{y\in\Omega} \frac{(p_t(x,y) - p_t(z,y))^2}{\phi_0(y)} \,. \tag{2}$$

This quantity is simply a weighted L^2 distance between the conditional probabilities $p_t(x,\cdot)$, and $p_t(z,\cdot)$. These probabilities can be thought of as features attached to the points x and z, and they measure the influence or interaction of these two nodes with the rest of the graph. If one increases t, one propagates the local or short-term influence of each node to its nearest neighbors, and this means that t also plays the role of a scale parameter. The comparison of these conditional probabilities introduces a notion of proximity that accounts for the connectivity of the points in the graph. In particular, unlike the shortest path, or geodesic distance, this metric is robust to noise as it involves an integration along all paths of length t starting from x or z.

The connection between the diffusion distance and the eigenvectors goes as follows (see appendix):

$$D_t^2(x,z) = \sum_{l\geq 1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \,. \tag{3}$$

Note that the ψ_0 does not appear in the sum because it is constant. This identity means that the right eigenvectors can be used to compute the diffusion distance. Furthermore, because of the spectrum decay, only a few terms are needed to achieve a given relative accuracy $\delta > 0$ in the previous sum. Let $m(t)$ be this number, and define the diffusion map

$$\Psi_t : x \longmapsto \begin{pmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{m(t)}^t \psi_{m(t)}(x) \end{pmatrix} . \tag{4}$$

This mapping provides coordinates on the data set Ω, and embeds the n data points into the Euclidean space $\mathbb{R}^{m(t)}$. In addition, the spectrum decay is the reason why dimension reduction can be achieved. This method constitutes a universal and data-driven way to represent a graph or any generic data set as a cloud of points in a Euclidean space. We also obtain a complete parametrization of the data that captures relevant modes of variability. Moreover, the dimension $m(t)$ of the new representation only depends on the properties of the random walk on the data, and not on the number of features of the original representation of the data. In particular, if we increase t, then $m(t)$ decreases and we capture the large scale geometry of the data.

B. Data merging using the Laplace-Beltrami normalization

We now direct our attention to the case when the original data points $\Omega = \{x_1, ..., x_n\}$ are assumed to approximately lie on a submanifold $\mathcal{M}$ of $\mathbb{R}^d$. The so called "manifold model" holds for a large variety of situations, such as when the data is produced by a source controlled by a few free parameters. For instance, consider the rotation of a human head and the lips motion of a speaker. We will study these examples later in this paper.

On the manifold $\mathcal{M}$, the data points were sampled with a density $q(\cdot)$ that may reflect some important aspect of the phenomenon that generated the data. For instance, as described in [12], for some data sets, the density is related to the free energy surface that governs the samples. On the other hand, the density may depend on the acquisition process and may be unrelated to intrinsic geometry or dynamics of the underlying phenomenon. In this situation, the distribution of the points is an artifact of the sampling process, and consequently, any "good" representation of the data should be invariant to the density.

Classical eigenmap methods provide an embedding that combines the information of both the density and geometry. For instance, with the Laplacian eigenmaps [2], one starts by forming the graph with Gaussian weights $w_\varepsilon(x,y) = \exp(-\|x-y\|^2/\varepsilon)$, and then constructs the random walk as described in the previous section. The eigenvectors are then used to embed the data set into a Euclidean space. It was shown in [10] that in the large sample limit $n \to +\infty$ and small scale $\varepsilon \to 0$, the eigenvectors tend to those of the Schrödinger operator $\Delta + E$, where Δ is the Laplace-Beltrami operator on $\mathcal{M}$, and E is a scalar potential that depends on the density q. As a consequence, the Laplacian eigenmaps representation of the data heavily depends on the density of the data points. In particular, it makes it impossible to fuse two data sets obtained from the same sensors but with different densities.

In order to solve this problem, we suggest to renormalize the Gaussian edge weights $w_\varepsilon(\cdot,\cdot)$ with an estimate of the density and to form the random walk on this new graph. This is summarized in Algorithm 1.

Let P_ε be the transition matrix with entries $p_\varepsilon(\cdot,\cdot)$. The asymptotics for P_ε are given in the following theorem.

Theorem 1: In the limit of large sample and small scales, we have

$$\lim_{\varepsilon \to 0} \lim_{n \to +\infty} \frac{I - P_\varepsilon}{\varepsilon} = \Delta \,.$$

In particular, the eigenvectors of P_ε tend to those of the Laplace-Beltrami operator on $\mathcal{M}$. We refer to [5] for a proof. This result shows that the diffusion embedding that one obtains from an appropriately renormalized Gaussian kernel does not depend on the density q of the data points of $\mathcal{M}$. This algorithm allows to successfully capture the nonlinear constraints governing the data, independently from the distribution of the points. In other words, it separates the geometry of the manifold from the density.

Algorithm 1 Approximation of the Laplace-Beltrami diffusion

```
Start with a rotation-invariant kernel $w_\varepsilon(x,y) = h\left(\frac{\|x-y\|^2}{\varepsilon}\right)$.
Let
    $q_\varepsilon(x) \triangleq \sum_{y\in\Omega} w_\varepsilon(x,y)\,,$
  and form the new kernel
    $\widetilde{w}_\varepsilon(x,y) = \frac{w_\varepsilon(x,y)}{q_\varepsilon(x)q_\varepsilon(y)}\,. \qquad (5)$
Apply the normalized graph Laplacian construction to this kernel, i.e., set
    $d_\varepsilon(x) = \sum_{z\in\Omega} \widetilde{w}_\varepsilon(x,y)\,,$
  and define the anisotropic transition kernel
    $p_\varepsilon(x,y) = \frac{\widetilde{w}_\varepsilon(x,y)}{d_\varepsilon(x)}\,.$
```

C. Out-of-sample extension and the geometric harmonics

In most applications, it is essential to be able to extend the low dimensional representation computed on a training set to new samples. Let Ω be a data set and Ψ_t be its diffusion embedding map. We now present the geometric harmonic scheme that allows us to extend Ψ_t to a new data set $\widetilde{\Omega}$. Since we need to relate the new samples to the training set, we will assume that Ω is a subset of a Euclidean space $\mathbb{R}^d$.

The focal point of our extension scheme is the distinction between the embedding kernel $\widetilde{w}_\varepsilon$ used to compute Ψ_t on Ω and the extension kernel k_σ used to extend Ψ_t onto the new data set $\widetilde{\Omega}$. It was shown in [9] that the properties required for the expansion kernel k_σ are significantly different than the ones of $\widetilde{w}_\varepsilon$ and somewhat contradicting. In particular, while computing Ψ_t one strives to use as small a scale $\sqrt{\varepsilon}$ as possible, while for the expanding kernel k_σ one would use a scale factor σ as large as possible. The geometric harmonic algorithm was first introduced in [9] and is based on the idea of using the Nyström extension to expand the eigenvectors of the specially designed kernel k_σ from Ω to $\widetilde{\Omega}$. These eigenvectors form a basis that can be used to extend any function f given on Ω to $\widetilde{\Omega}$ and in particular the vector function Ψ_t. In our application we used a Gaussian extension kernel $k_\sigma(x,y) = e^{-\|x-y\|^2/\sigma^2}$ to extend Ψ_t computed by the Laplace-Beltrami kernel given in Equation 5. In previous works [7], the Nyström extension was used to extend Ψ_t using the same kernel $\widetilde{w}_\varepsilon$.

Next, we discuss the design of the extension kernel k_σ and provide a scheme for its computation in Algorithm 2. The design is based on finding an equilibrium between σ, the width of the extension kernel k_σ and the reconstruction error of the function f on Ω using only a subset of the eigenvectors of k_σ. On the one hand, we aim to increase σ as much as possible to maximize the extension range. But on the other hand, as shown below, this also increases the reconstruction error of f. Hence, the reconstruction error limits the maximal extension range. In fact, this limitation can be regarded as relating the complexity of the function on the training set to the distance to which it can be extended off this set. Here, the notion of complexity is measured in terms of frequency content on the training domain. For instance, a constant function has almost no complexity and one should be able to extend it in the entire space. If the number of oscillations of this function increases, then the distance to which one can extend it gets smaller.

We first recall the idea of Nyström extension [16]. Let $\sigma > 0$ be a scale of extension, and consider the eigenvectors and eigenvalues of a Gaussian kernel of width σ on the training set Ω:

$$\mu_l \varphi_l(x) = \sum_{y\in\Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \Omega\,.$$

Since the kernel can be evaluated the entire space, it is possible to take any $x \in \mathbb{R}^d$ in the right-hand side of this identity. This yields the following definition of the Nyström extension of φ_j from Ω to $\mathbb{R}^d$:

$$\overline{\varphi}_l(x) \triangleq \frac{1}{\mu_l} \sum_{y\in\Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \mathbb{R}^d\,. \tag{6}$$

Numerically, we have extended φ_l only to a distance σ from Ω. Such an extension is termed "geometric harmonic". As the eigenvectors φ_l form an orthonormal set, an arbitrary function f can be extended from Ω to R^d by expressing it as a linear combinations of the geometric harmonics φ_l.

However, as it can be seen from Equation 6 and from the fact that $\mu_l \to 0$, the extension of some φ_l's is an ill-posed linear operation. Indeed, the extension of the first $(l+1)$ eigenfunctions of the Gaussian kernel has a condition number equal to μ_0/μ_l. The only way to control the conditioning of this procedure is to perform regularization by retaining only the coefficients for which $\mu_0/\mu_l < \eta$ (where η is a bound on the condition number that plays the role of a regularization parameter):

$$f \simeq \sum_{l:\, \mu_0 < \eta \mu_l} \langle \varphi_l, f \rangle \varphi_l \,.$$

This approximation generates an error of reconstructing f on Ω. Therefore if we fix an admissible error threshold $\tau > 0$, and check whether this error is smaller or larger than τ. In the former case, the function f has a low-frequency content and can safely be extended at scale σ. In the latter case, a non-negligible energy is lost in high frequency coefficients, and f cannot be extended at scale σ. Consequently, the scale σ has to be reduced. A smaller σ results in a slower decay of the eigenvalues μ_l and an improved condition number η. These observations give rise to the multiscale extension scheme summarized in Algorithm 2.

D. Multi-cue alignment and data matching

The purpose of this section is to explain how the diffusion embedding can be efficiently used for data matching. Suppose that one has two data sets $\Omega_1 = \{x_1, ..., x_n\}$ and $\Omega_2 = \{y_1, ..., y_{n'}\}$ for which one would like to find a correspondence, or detect similar patterns and trends, or on the contrary, underline their dissimilarity and detect anomalies. This type of task is very common in applications related to marketing, fraud detection or even counter-terrorism. However, working with the data in its original form can be quite difficult as the two sets typically consist of measurements of very different nature. For instance Ω_1 could be a collection of measurements related to wether in a given region, whereas Ω_2 could describe agriculture production in the same region. As a consequence, it is almost always impossible to directly compare the two data sets. The main idea that we introduce here is that the diffusion maps provide a canonical representation of data sets. This new representation is based on the graph structure of a set, which is often the relevant structure in the context of data matching. As a consequence, instead of comparing the sets in their original forms, it can be much more efficient to compare their embeddings. In particular, if Ω_1 and Ω_2 are expected to have similar structures, then they should have similar embeddings.

Algorithm 2 Multiscale extension scheme of diffusion coordinates via geometric harmonics

```
1: Let $\Omega \subset \mathbb{R}^d$ be the training set and $\psi_i : \Omega \to \mathbb{R}$ be the diffusion coordinate to be extended ($1 \leq i \leq m(t)$). Choose a condition number $\eta > 0$ and an admissible error $\tau > 0$.
2: Choose an initial (large) scale of extension $\sigma = \sigma_0$.
3: Compute the eigenfunctions of the Gaussian kernel with width $\sigma$ on the training set $\Omega$:
       $\mu_l \varphi_l(x) = \sum_{y \in \Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \Omega\,,$
   and expand $f$ on this orthonormal basis (on the training set $\Omega$):
       $f(x) = \sum_{l \geq 0} c_l \varphi_l(x) \text{ where } x \in \Omega\,.$
4: Compute the error of reconstruction on the training set that one obtains by retaining only the coefficients such that $\eta > \mu_0/\mu_l$ in the sum above:
       $Err = \left( \sum_{l:\, \eta \leq \mu_0/\mu_l} |c_l|^2 \right)^{\frac{1}{2}} .$
   If $Err > \tau$ then divide $\sigma$ by 2 and go back to point 3. Otherwise continue.
5: For each $l$ such that $\eta > \mu_0/\mu_l$, extend $\varphi_l$ via the Nyström procedure:
       $\overline{\varphi}_l(x) \triangleq \frac{1}{\mu_l} \sum_{y \in \Omega} e^{-\|x-y\|^2/\sigma^2} \varphi_l(y) \text{ where } x \in \mathbb{R}^d\,,$
   and define the extension $\overline{f}$ of $f$ to be
       $\overline{f}(x) \triangleq \sum_{l \geq 0} c_l \overline{\varphi}_l(x) \text{ where } x \in \mathbb{R}^d\,.$
```

We illustrate these ideas by presenting a semi-supervised algorithm for finding a one-to-one correspondence between two data sets. The scheme we introduce consists in aligning two graphs in a nonlinear fashion, based on a finite number of landmarks. More precisely, suppose that we have $k < n, n'$ landmarks in each set, that is a sequence of k pairs $(x_{\sigma(1)}, y_{\tau(1)}), ..., (x_{\sigma(k)}, y_{\tau(k)})$ for which there is a known correspondence. This set of examples is the only prior information we use in the algorithm. We assume that

$x_{\sigma(1)} \neq x_{\sigma(2)} \neq ... \neq x_{\sigma(k)}$. The scheme given in Algorithm 3 computes a surjective function $g : \Omega_1 \rightarrow \Omega_2$ such that $g(x_{\sigma(1)}) = y_{\tau(1)}, ..., g(x_{\sigma(k)}) = y_{\tau(k)}$.

Algorithm 3 Nonlinear graph alignment

```
Start with $k$ landmarks $(x_{\sigma(1)}, y_{\tau(1)}), ..., (x_{\sigma(k)}, y_{\tau(k)})$.
Compute the diffusion embeddings $\{\Phi_t(x_1), ..., \Phi_t(x_n)\}$ and $\{\Phi_t(y_1), ..., \Phi_t(y_{n'})\}$ of $\Omega_1$ and $\Omega_2$ where
    $t$ is chosen so that at least $k-1$ eigenvectors are retained.
Compute the affine function $f : \mathbb{R}^{k-1} \rightarrow \mathbb{R}^{k-1}$ that satisfies
    $f(x_{\sigma(1)}) = y_{\tau(1)}, ..., f(x_{\sigma(k)}) = y_{\tau(k)}$ .
Define the correspondence between $\Omega_1$ and $\Omega_2$ by
    $g(x_i) = \arg\min_{y \in \Omega_2} \{\|f(x_i) - y\|\}$ ,
    where $x_i \in \Omega_1$,
```

The number of eigenvectors used for the alignment is directly related to the number of landmarks, which in turns, represents the quantity of prior information for aligning. The larger the number of known constraints on the alignment, the larger the dimensionality of the aligning mapping. This observation is consistent with the fact that higher order eigenvectors capture finer structures. Note also that the linear function that we use for aligning induces a nonlinear mapping defined on lower dimensional embedding of the sets. These observations pave the way for a general sampling theory for data sets. Indeed, the landmarks can be regarded as forming a subsampling of the original data sets. This subset determines the largest (or Nyquist) frequency used to represent the original set. This frequency is measured as the number of eigenvectors used.

III. Experimental results

A. Application to lip-reading

The validity of our approach is now demonstrated by applying it to lip reading and sequence alignment, which are typical high-dimensional data analysis problems. In particular, lip reading has gained significant attention [17], [18], [19], [20], [21] and we now provide background and previous results in that field. The ultimate goal of lip reading is to design human-like man-machine interfaces allowing automatic comprehension of speech, which in the absence of sound is denoted as lip-reading and the synthesis of realistic lip movement. The design of such a system involves three main challenges: first, the feature extraction, which aims at converting the images of the lips into a useful description, must be achieved with minimal preprocessing. Then, in order to be efficiently processed, the data must be transformed via a dimension reduction technique. Last, in order to assimilate new data for recognition, one must be able to perform data fusion.

Previous schemes have mainly focused on the first two points. Concerning the feature extraction, some works [17], [21] analyze directly the intensity values of the input images, while others [22], [18] start by detecting curves and points of interest around the mouth whose locations are then used as features. The combination of audio-visual cues was used in [23] where the visual cues are the extracted lip contours which are tracked over time. We note that combining audio-visual is beyond the scope of this work and will be dealt by us in the future. Identifying, tracking and segmenting the lips is a difficult task and possible solutions include: active contours [24], probabilistic models [25] and the combination of multiple visual cues (shape, color and motion) [26] to name a few. In practice, one strives to use a simple preprocessing scheme as possible and in our scheme we employ a simple stabilization scheme discussed below.

Regarding the dimensionality reduction, several schemes have been used. Preliminary work employed linear algorithms such as the PCA and SVD subspace projections [22], [21]. For instance, Li *et al* [21] use a linear PCA scheme similar to the eigenfaces approach to face detection. Recognition is performed by correlating an input sequence with the eigenfeatures obtained from PCA. More recent schemes [17] utilize non-linear approaches such as the MDS [27]. Some of the techniques provide a general embedding framework for lipreading analysis [17], while others [21], [18] concentrate on a particular task such as phoneme or word identification. The work in [28] is of particular interest, since it is one of the first to explicitly formulate the lipreading problem as a “Manifold Learning” issue and tries to derive the inherent constraints embedded in the space of lip configurations. A Hidden Markov Model (HMM) is used to model a small number of words (names of four drinks) which define the Markov states and the manifold. The HMM is then used to recognize the drinks’ names where the input is given by tracking the outer lips contour using Active Contours. Utilizing both audio and visual information significantly decreased the error rate, especially in noisy environments. Kimmel and Aharon [17] applied the MDS scheme to visual lips representation, analysis and synthesis. A set of lips images is aligned and embedded in a two dimensional domain which is then sampled uniformly in the embedding domain to achieve uniform density. The pronunciation of each word is defined as a path over the embedding domain and used for visual speech recognition, by path matching. Lips motion synthesis is derived by computing the geodesic path over the embedding domain, where the start and end point are given as input.

Analysis of lip data constitutes an application where it is important to separate the set of nonlinear constraints on the data from the distribution of the points. As an illustration of the Laplace-Beltrami normalization as well as the out-of-sample extension scheme, we now describe an elementary experiment that paves the way to building automatic lip-reading machines, and more generally, machine learning systems.

We recorded a movie of the lips of a subject reading a text in English. The subject was then asked to repeat each digit “zero”, “one”, ... , “nine” 40 times. A minimal preprocessing was applied to the recorded sequence. More precisely, it was first converted from colors to gray level. Moreover, using a marker put at the tip of the nose of the speaker during the recording, we were able to automatically crop each frame into a rectangular area around the lips. Each of these new frames was then regarded as a point in $\mathbb{R}^{140\times110}$, where 140×110 is the size of the cropped area.

The first part of the data sets, consisting of approximately 5000 frames, corresponds to the speaker reading the text. These points were used to form a graph with Gaussian weights $\exp(\|x_i - x_j\|^2/\varepsilon)$ on the edges, for an appropriately chosen scale $\varepsilon > 0$. The distance $\|x_i - x_j\|$ was merely calculated as the Euclidean L^2 distance between frames i and j. We then renormalized the Gaussian weights using the Laplace-Beltrami normalization described in Section II-B. By doing so, our analysis focused on viewing the mouth as a constrained mechanical system. In order to obtain a low-dimensional parametrization of these nonlinear constraints, we computed the diffusion coordinates on this new graph. The embedding in the first 3 eigenfunctions is shown on Figure 1.

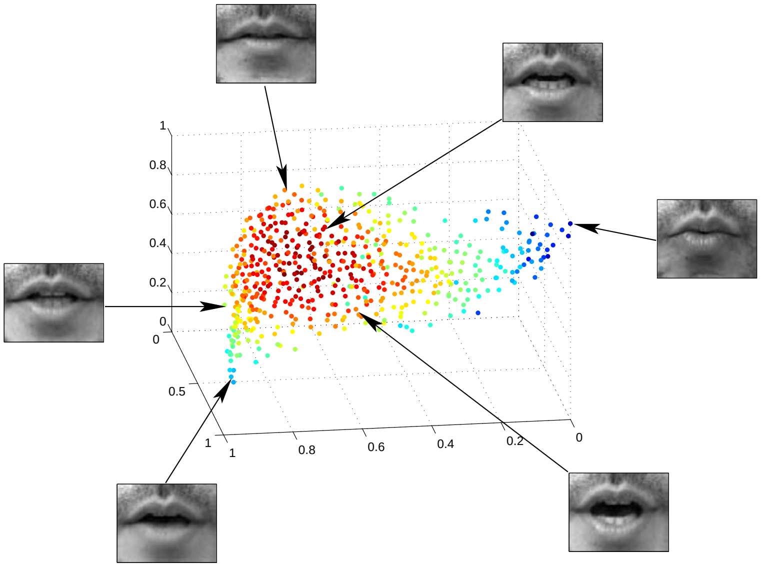


Fig. 1. The embedding of the lip data into the top 3 diffusion coordinates. These coordinates essentially capture two parameters: one controlling the opening of the mouth and the other measuring the portion of teeth that are visible.

The task we wanted to perform was word recognition on a small vocabulary. The example that we considered was that of identification of digits. Each word “zero”, “one”,..., “nine” is typically a sequence 25 to 40 frames that we need to project in the diffusion space. In order to do so, we used the geometric harmonic extension scheme presented in Section II-C to extend each diffusion coordinate to the frames corresponding to the subject pronouncing the different digits. After this projection, each word can be viewed as a trajectory in the diffusion space. The word recognition problem now amounts to identifying trajectories in the diffusion space.

We can now build a classifier based on comparing a new trajectory to a collection of labelled trajectories in a training set. We randomly selected 20 instances of each digit to form a training set, the remaining 20 being used as a testing set. In order to compare trajectories in the diffusion space, a metric is needed, and we chose to use the Hausdorff distance between two sets Γ_1 and Γ_2, defined as

$$d_H(\Gamma_1, \Gamma_2) = \max\left\{ \max_{x_2 \in \Gamma_2} \min_{x_1 \in \Gamma_1} \{\|x_1 - x_2\|\}, \max_{x_1 \in \Gamma_1} \min_{x_2 \in \Gamma_2} \{\|x_1 - x_2\|\} \right\} .$$

Although this distance does not use the temporal information, it has the advantage of not being sensitive to the choice of a parametrization or to the sampling density for either set Γ_1 and Γ_2. For a given trajectory Γ from the testing set, our classifier is a nearest-neighbor classifier for this metric, *i.e.*, the class of Γ is decided to be that of the nearest trajectory (for d_H) in the training set. The performance of this classifier averaged over 100 random trials is shown in Table I. In this case, the data set was embedded in 15 dimensions.

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	0.93	0	0	0.01	0	0	0.06	0	0	0
one	0	1	0	0	0	0	0	0	0	0
two	0.05	0	0.88	0.05	0.01	0	0.01	0	0	0
three	0.01	0	0.02	0.93	0	0	0.01	0.01	0.01	0.01
four	0	0	0.01	0.01	0.97	0	0	0.01	0	0
five	0	0	0	0.01	0	0.84	0.01	0.14	0	0.01
six	0.04	0	0	0.01	0	0	0.92	0.02	0	0.01
seven	0.02	0	0	0.04	0	0.07	0.10	0.69	0.05	0.03
eight	0	0.01	0	0	0	0.03	0.01	0.04	0.77	0.14
nine	0	0	0	0.02	0	0	0	0.02	0.12	0.85

TABLE I

CLASSIFIER PERFORMANCE OVER 100 RANDOM TRIALS. EACH ROW CORRESPONDS THE CLASSIFICATION DISTRIBUTION OF A GIVEN DIGIT OVER THEN 10 CLASSES. THE DATA SET WAS EMBEDDED IN 15 DIMENSIONS.

The classification error ranges from 0% to 31% with an average of 12.2%. The best classification rate is achieved for the word “one” which, in terms of visual information, stands far away from the other digits. In particular, typical sequences of “one” involve frames with a round open mouth, with no teeth visible (see first row of Figure 2). These frames essentially never appear for other digits. The worst classification job is for the word “seven” which seems to be highly confused with the words “five” and “six”. As shown on Figure 2, typical instances of these words appear to be similar in that the central frames involve an open mouth with visible teeth. In the case of the “six” and “seven”, teeth from the lower jaws are visible because of the “s” sound. Regarding the similarity between “five” and “seven”, the ”f” and ”v” sounds translate into the lower lip touching the teeth of the upper jaw.

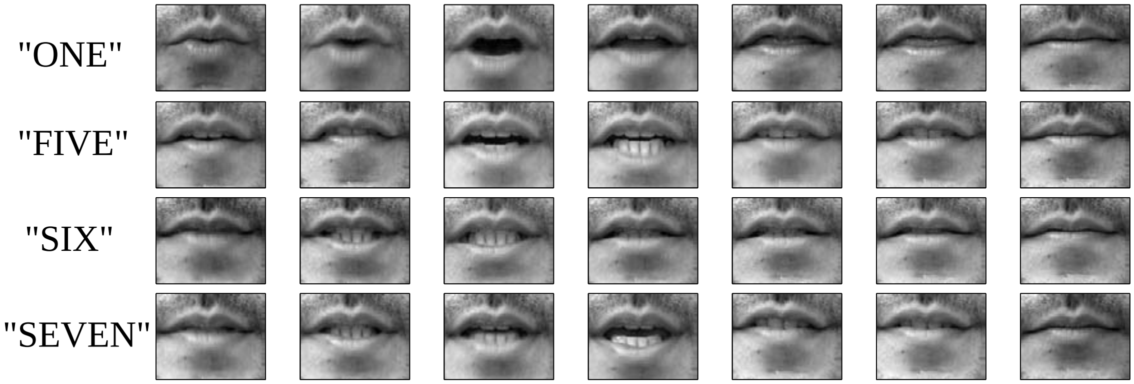


Fig. 2. Typical frames for the words “one”, “five”, “six”, “seven”.

B. Synchronization of head movement data

We now illustrate the concept of graph alignment as well as the algorithm presented in Section II-D. We recorded 3 movies of subjects wearing successively a yellow, red and black mask. Each subject was asked to move their head in front of the camcorder. We then considered the three sets consisting of all frames of each movie. Let YELLOW, RED and BLACK denote these sets. Our goal was to synchronize the movements of the different masks by aligning the 3 diffusion embeddings. It is to be noted that working directly in image space would be highly inefficient since any picture of the red or black mask is at a large distance from the set of pictures of the yellow mask. On the contrary, the diffusion coordinates will capture the intrinsic organization of each data sets, and therefore will provide a canonical representation of the sets that can be used for matching the data.

Each set of frames was regarded as a collection of points in $\mathbb{R}^{10000}$, where the dimensionality coincides with the number of pixel per image. Following the lines of our algorithm, we formed a graph from each set with Gaussian weights $\exp(\|x_i - x_j\|^2/\varepsilon)$, for an appropriately chosen scale $\varepsilon > 0$. Here $\|x_i - x_j\|$ represents the L^2 norm between images i and j. We expect each set to lie approximately on a manifold of dimension 2, as each subject essentially moved their head along two angles α and β shown on Figure 3. and as the light conditions were kept the same during the recording.

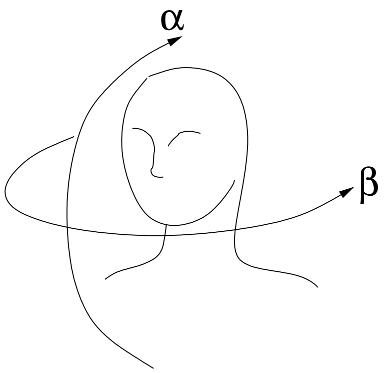
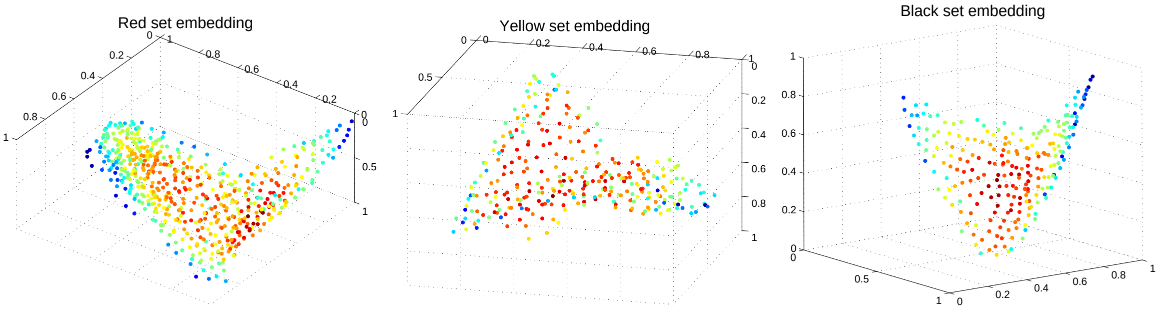


Fig. 3. Each subject essentially moved their head along the two angles α and β. There was almost no tilting of the head. Hence, the data points approximately lie on a submanifold of dimension 2.

It is clear that the density of points on this manifold is essentially arbitrary and varies with each subject and recording. Since we were only interested in the space of constraints, that is the geometry of the manifold, we renormalized the Gaussian weights according to the algorithm described in Section II-B, and constructed a Markov chain that approximates the Laplace-Beltrami diffusion. We then defined 8 matching triplets of landmarks in each set. The landmarks were chosen to correspond to the main head positions. We computed the diffusion embedding in 7 dimensions and we then calculated two affine functions $g_{YR} : \mathbb{R}^7 \to \mathbb{R}^7$ and $g_{YB} : \mathbb{R}^7 \to \mathbb{R}^7$ that match the landmarks from YELLOW to BLACK, and from YELLOW to RED.

Two conclusions can be drawn from this experiment. First, the diffusion embedding revealed that the ata sets were approximately 2-dimensional, as expected (see Figure 4 for the embeddings in the first 3 diffusion coordinates). The diffusion coordinate captured the main parameters of variability, namely the angles α and β. Second, the two functions g_{YB} and g_{YR} allowed us to drive the movements of the black and red masks from those of the yellow mask. The result of the matching of the three data sets is shown on Figure 5.



(a)

(b)

(c)

Fig. 4. The embedding of each set in the first 3 diffusion coordinates. The color encodes the density of points.

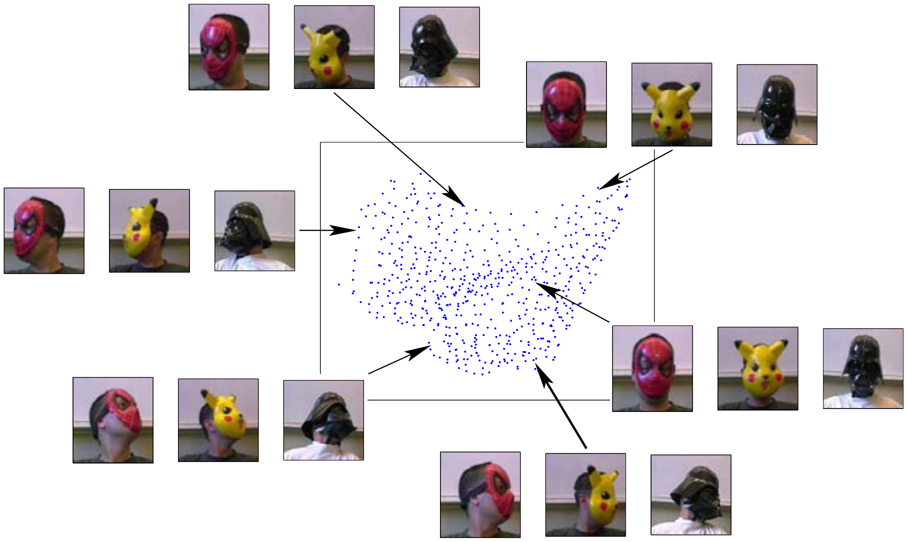


Fig. 5. The embedding of the YELLOW set in three diffusion coordinates and the various corresponding images after alignment of the RED and BLACK graphs to YELLOW.

IV. Conclusion and future work

In this work we introduced diffusion techniques as a framework for data fusion and multi-cue data matching by addressing several key issues. First, we underlined the importance of the Laplace-Beltrami normalization for data fusion by showing that it allows to merge data sets produced by the same source but with different densities. In particular, the Laplace-Beltrami embedding provides a canonical, density invariant embedding which is essential for data matching. For example, matching the visual data of different speakers and the "rotating heads" sequence. Second, we suggested a new data fusion scheme, by extending spectral embeddings using the geometric harmonics framework. Finally, we presented a spectral graph alignment approach to data fusion.

Our scheme was successfully applied to lip reading where we achieved high accuracy with minimal preprocessing. We also demonstrated the alignment of high dimensional visual data ("rotating heads" sequence).

In the future we intend to extend our approach to multi-cue data analysis, by integrating different features in a multigraph, constructed by combining the graphs of the different features over the data set. Finally, we are studying a spectral based approach to the analysis of signals as Markov random processes. Our current work did not utilize the temporal information of the video sequences, whose frames were considered as samples of a random variable. By constructing a Markov process model, we intend to improve the lips reading accuracy using the Viterbi algorithm.

V. Acknowledgments

The authors would like to thank Steven Zucker for helpful discussions during the course of this work.

Appendix

Diffusion distance and eigenfunctions

The random walk constructed from a graph via the normalized graph Laplacian procedure yields a Markov matrix P with entries $p_1(x, y)$. As it is well known [13], this matrix is in fact conjugate to a symmetric matrix A with entries $a(x, y)$, given by

$$a(x, y) = \sqrt{\frac{d(x)}{d(y)}} p_1(x, y) = \frac{w(x, y)}{\sqrt{d(x)d(y)}} .$$

Therefore A has n eigenvalues $\lambda_0, ..., \lambda_{n-1}$ and orthonormal eigenvectors $v_0, ..., v_{n-1}$. In particular,

$$a(x, y) = \sum_{l=0}^{n-1} \lambda_l v_l(x) v_l(y) . \tag{7}$$

This implies that P has the same n eigenvalues. In addition, it has n left eigenvectors $\phi_0, ..., \phi_{n-1}$ and n right eigenvectors $\psi_0, ..., \psi_{n-1}$. Also, it can be checked that

$$\phi_l(y) = v_l(y) v_0(y) \text{ and } \psi_l(x) = v_l(x)/v_0(x) . \tag{8}$$

Furthermore, it can be verified that $v_0(x) = \sqrt{d(x)}$, and therefore $\phi_0(y) = d(y)$ and $\psi_0(x) = 1$. In addition,

$$\phi_0(x)\psi_l(x) = \phi_l(x) . \tag{9}$$

It results from Equations 7 and 8 that P^t admits the following spectral decomposition:

$$p_t(x, y) = \sum_{l=0}^{n-1} \lambda_l^t \psi_l(x) \phi_l(y) , \tag{10}$$

together with the biorthogonality relation

$$\sum_{y \in \Omega} \phi_i(y) \psi_j(y) = \delta_{ij} , \tag{11}$$

where δ_{ij} is Kronecker symbol. Combining this last identity with Equation 9, one obtains

$$\sum_{y \in \Omega} \frac{\phi_i(y)\phi_j(y)}{\phi_0(y)} = \delta_{ij} .$$

This means that the system $\{\phi_l\}$ is orthonormal in $L^2(\Omega, 1/\phi_0)$. Therefore, if one fixes x, Equation 10 can interpreted as the decomposition of the function $p_t(x, \cdot)$ over this system, where the coefficients of decomposition are $\{\lambda_l^t \psi_l(x)\}$.

Now by definition,

$$D_t(x, z)^2 = \sum_{y \in \Omega} \frac{(p_t(x, y) - p_t(z, y))^2}{\phi_0(y)} = \|p_t(x, \cdot) - p_t(z, \cdot)\|^2_{L^2(\Omega, 1/\phi_0)} .$$

Therefore,

$$D_t(x,y)^2 = \sum_{l=0}^{n-1} \lambda_l^{2t} (\psi_l(x) - \psi_l(z))^2 \, .$$

References

[1] S. Roweis and L. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, pp. 2323–2326, 2000.

[2] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 6, no. 15, pp. 1373–1396, June 2003.

[3] D. Donoho and C. Grimes, “Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[4] Z. Zhang and H. Zha, “Principal manifolds and nonlinear dimension reduction via local tangent space alignement,” Department of computer science and engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.

[5] R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[6] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: a geometric framework for learning from examples,” University of Chicago, Tech. Rep. TR-2004-06, 2004.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the nyström method.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[8] Y. Bengio, J.-F. Paiement, and P. Vincent, “Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering,” Université de Montréal, Tech. Rep. 1238, 2003.

[9] R. Coifman and S. Lafon, “Geometric harmonics,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[10] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[11] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, “Geometric diffusions as a tool for harmonics analysis and structure definition of data: Multiscale methods,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7432–7437, May 2005.

[12] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, “Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[13] F. Chung, *Spectral graph theory*. CBMS-AMS, May 1997, no. 92.

[14] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Tran PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[15] M. Meila and J. Shi, “A random walk’s view of spectral segmentation,” *AI and Statistics (AISTATS)*, 2001.

[16] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University, 1988.

[17] M. Aharon and R. Kimmel, “Representation analysis and synthesis of lip images using dimensionality reduction,” Technion, Tech. Rep. TR: CIS-2004-01, 2004.

[18] B. Christoph, C. Michele, and S. Malcolm, “Video rewrite: driving visual speech with audio,” in *SIGGRAPH ’97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 353–360.

[19] E. Cosatto and H. P. Graf, “Photo-realistic talking-heads from image samples.” *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 152–163, 2000.

[20] C. Bregler, S. Manke, and H. Hild, “Improving connected letter recognition by lipreading,” in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Processing*, 1993.

[21] N. Dettmer and M. Shah, “Visually recognizing speech using eigensequences,” *Computational Imaging and Vision*, pp. 345–371, 1997.

[22] I. Matthews, T. Cootes, A. Bangham, S. Cox, and R. Harvey, “Extraction of visual features for lipreading.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 198–213, 2002.

[23] A. V. Nefian, L. H. Liang, X. X. Liu, X. Pi, and K. Murphy, “Dynamic bayesian networks for audio-visual speech recognition,” *Journal of Applied Signal Processing,*, vol. 2002, no. 11,, pp. 1274–1288, 2002.

[24] J. Luettin, N. A. Thacker, and S. W. Beet, “Active shape models for visual speech feature extraction,” in *Speechreading by Humans and Machines*, ser. NATO ASI Series, Series F: Computer and Systems Sciences, D. G. Storck and M. E. H. (editors), Eds. Berlin: Springer Verlag, 1996, vol. 150, pp. 383–390.

[25] J. Luettin and N. A. Thacker, “Speechreading using probabilistic models,” *Computer Vision and Image Understanding*, vol. 65, no. 02, pp. 163–178, 1997.

[26] Y.-L. Tian, T. Kanade, and J. Cohn, “Robust lip tracking by combining shape, color and motion,” in *Proceedings of the 4th Asian Conference on Computer Vision (ACCV’00)*, January 2000.

[27] I. Borg and P. Groenen, *Modern Multidimensional Scaling - Theory and Applications*. Springer-Verlag New York Inc., 1997.

[28] C. Bregler, S.Omohundro, M.Covell, M.Slaney, S.Ahmad, D.A.Forsyth, and J.A.Feldman, “Probabilistic models of verbal and body gestures,” in *Computer Vision in Man-Machine Interfaces*, R. Cipolla and A. eds, Eds. Cambridge University Press, 1998.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Sensor fusion by diffusion maps

Ronald R. Coifman, Yosi Keller, Stéphane Lafon and Steven W. Zucker

Program of Applied Mathematics, Department of Mathematics,

Yale University New Haven, CT 06520.

yosi.keller@yale.edu,stephane.lafon@gmail.com,steven.zucker@yale.edu

Abstract

Data fusion and the analysis of high-dimensional multisensor data, are fundamental tasks in many research communities. In this work we propose a unified embedding scheme for multi sensory data, which is based on the recently introduced diffusion framework. Our scheme is purely data-driven and assume no a-priory knowledge of the underlying statistical or deterministic models. Our approach is based on embedding separately each of the input channels and combining the resulting diffusion coordinates. In particular we use the density invariant Laplace-Beltrami embedding. In order to verify the efficiency of our approach, we apply it to typical multisensory statistical learning and clustering applications, such as spoken-digit recognition and multi-cue image segmentation. For both applications we experimentally show that using the unified multisensor embedding, allows better performance than the one achieved by any single sensor.

1 Introduction

The first task performed by any data processing system is data acquisition or sampling, in which measurements are collected through a number of sensors. In this work, we refer to a “sensor” any information stream produced by an acquisition device or, more generally, any descriptor used to represent some form of data. Single-sensor systems, which process data coming from a unique information channel, have been successfully used in various context ranging from object recognition (e.g. Sonar) to the medical area (e.g. blood pressure sensors). However, it was early recognized that these systems typically suffer from incompleteness due to the fact that a single sensor is almost never sufficient to capture all of the relevant information related to a phenomenon. For instance, in medical imaging different sensors, such as X-Ray, CT, MRI and others, asses different physical properties. This issue was further studied in the context of remote-sensing (SAR, FLIR, IR and optical sensors). In particular, different sensors are subject to different limitations restricting their usability. For example, in remote sensing, optical sensors have significantly better resolution and lower SNR than Radar based SAR sensors, yet SAR sensors are immune to atmospheric conditions and can be used in any weather conditions. The multisensor approach allows to resolve ambiguities and reduce uncertainties that may arise in some situations, such as object recognition. For example, consider the work by Kidron et. al. [1] who detected image pixels within a video sequence that were related to the creation of sound, given the visual and audio data. Using only the visual data was insufficient as some of the motions in a scene were unrelated to the sound creation.

Note also that many living species rely heavily on a multisensor approach (most humans can see, hear, taste...). In particular, the fusion of audio-visual cues was shown to enhance perception [2, 3]. Last, it is often more cost-efficient to combine a variety of cheap sensors rather than to deal with an expensive single sensor.

The use of high-dimensional multisensor signals requires several tasks. First, the signals have to be embedded in a low-dimensional space that recovers the underlying manifold. When the different data sources are not synchronized and have to be aligned, this manifold can also be used for alignment [4]. In particular, as different sources might sample the same phenomenon with different densities, the alignment requires a density-invariant embedding, as eigenmap representations [5, 6, 7, 8] depend on the density of the points on the underlying manifold.

A second task is the alignment and synchronization of different multisensor sources. This was extensively studied in the remote sensing and medical imaging communities. In such applications, due to the different physical characteristics of various imaging sensors, the relationship between the intensities of matching pixels is often complex and unknown a priori. The common approach to multisensor image alignment is to compute canonical representations of image features, which are invariant to the dissimilarity between the different sensors and capture the essence of the image. Theses representations include geometrical primitives such as feature points, contours and corners [9, 10, 11]. Such approaches apply a deterministic a-priori know model that relates the measurements of the different input channels.

A general purpose approach to high-dimensional data embedding and alignment was presented by Ham et. al [12]. Given a set of a-priori corresponding points in the different input channels, a constrained formulation of the graph Laplacian embedding is derived. First, they add a term fixing the embedding coordinates of certain samples to predefined values. Both sets are then embedded separately, where certain samples in each set are mapped to the same embedding coordinates. Second, they describe a dual embedding scheme, where the constrained embeddings of both sets are computed simultaneously, and the embeddings of certain points in both datasets are constrained to be identical.

Kidron et.al [1] applied canonical correlation analysis to multisensor event detection. Their approach uses a parametric form of the covariance matrices to compute maximally correlated one-dimensional embeddings of the audio and video input signals. A sparsity constraint was applied to regularize the otherwise underconstrained embedding problem, where the constraint corresponds to the sparsity of the detected events.

There is also a large body of literature in engineering related to multisensor integration. These approaches can be classified into three categories [13]. First, some techniques are based on physical models for the data, like in the case of Kalman filtering. Another category corresponds to methods employing a parametric model for the data or the sensors. For instance this is the case of Bayesian inference, of the Dempster-Shafer method or Neural Networks. These techniques usually exhibit a high sensitivity to the accuracy of these models [14]. The third group consists of cognitive-based methods, which aim at mimicking human inference. One of the main tools is fuzzy logic. But there again, one needs to specify subjective membership functions. It therefore appears that many of these techniques rely on prior information.

A problem related to data fusion is the fusion of multiple partitionings [15]. The focal point there is to fuse together different *partitionings*, rather than different data *sources* as in the general data fusing problem. This approach boils down to embedding the data in a one-dimensional space (the partitions index). As this is not a metric space, a distance metric can be defined directly and the work in [15] uses the co-association matrix as a binary similarity measure.

A related problem was recently studied by the computer vision community in the context of multi-cue image segmentation. These works are of particular interest, as (similar to our approach) they are based on spectral embeddings [16]. In [17] Yu presents a segmentation scheme that integrates edges detected at multiple scales. These are shown to provide complementary segmentation cues. Given the affinity matrices computed using the edges at each scale, a simultaneous segmentation is computed using a novel criterion called average cuts. This approach does not explicitly assume the cues are multiscale and can be applied to using different cues rather than a single cue in different scales. Other works [18, 19], deal with the fusion of a single multiscale cue in images and can be applied directly to multisensor data

In this work we derive a unified low-dimensional representation, given as set of different input channels related to a particular phenomenon. We assume that the input signals are aligned and derive a unified representation of them, useful for statistical learning tasks and data partitioning. We compute a unified low-dimensional representation and show that it combines the information encoded in the different signals. Thus, is better able to parameterize complex phenomena. We start by computing low-dimensional embeddings of each of the input signals using the diffusion framework [20, 21] and for that we utilize the Laplace-Beltrami density invariant scheme [22]. The multisensor scheme is first applied to statistical learning by analyzing audio-visual based spoken-digit recognition and we compare our result to the results of the visual-only lip-reading given in [4]. Then we apply it to multi-cue image segmentation, where the multisensor data is related to different image cues: RGB, contours and texture. Compared to prior works, the presented approach does not require any deterministic model of the data or its statistics (covariance matrices etc.), and the structures that they recover are completely data-driven. In particular, we resolve the density-dependence issue of the embeddings that was largely overlooked in prior works.

This paper is organized as follows: We describe the foundations of the diffusion based embeddings and introduce the unified, fused multisensor embedding in Section 2. Our scheme is then experimentally verified in Section 3, while concluding remarks and future extensions are discussed in Section 4.

2 Multi-sensor integration

In this section we present the proposed data fusion scheme. We start by describing low-dimensional spectral embeddings and then extend them to derive the density-invariant Laplace-Beltrami embedding. A more detailed description can be found in [4], while the mathematical foundations are given in [22]. Given a set $\Omega = \{x_1, ..., x_n\}$ of data points, we start by constructing a weighted symmetric graph where each data point x_i corresponds to a node. Two nodes x_i and x_j are connected by an edge with weight $w(x_i, x_j) = w(x_j, x_i)$ reflecting the degree of similarity (or affinity) between these two points. The weight function $w(\cdot, \cdot)$ describes the first-order interaction between the data points and its choice is application-driven. For instance, in applications where a distance $d(\cdot, \cdot)$ already exists on the data, it is custom to weight the edge between x_i and x_j by $w(x_i, x_j) = \exp(-d(x_i, x_j)^2/\varepsilon)$, where $\varepsilon > 0$ is a scale parameter. In this paper, although our method would apply to general weights, we will mainly focus on this type of Gaussian-weight graph.

Following a classical construction in spectral graph theory [23] and in manifold learning **[24]**, namely the normalized graph Laplacian, we now create a random walk on the data set Ω by forming the kernel

$$p_1(x_i, x_j) = \frac{w(x_i, x_j)}{d(x_i)},$$

where $d(x_i) = \sum_{x_k \in \Omega} w(x_i, x_k)$ is the degree of node x_i. As we have that $p_1(x_i, x_j) \geq 0$ and $\sum_{j \in \Omega} p_1(x_i, x_j) = 1$, the quantity $p_1(x_i, x_j)$ can be interpreted as the probability of a random walker to jump from x_i to x_j in a single time step [23, 25]. If P is the $n \times n$ matrix of transition of this Markov chain, then taking powers of this matrix amounts to running the chain forward in time. Let $p_t(\cdot, \cdot)$ be the kernel corresponding to the t^{th} power of the matrix P. Then, $p_t(\cdot, \cdot)$ describes the probabilities of transition in t time steps. The essential point of the diffusion framework is the idea that running the chain forward will reveal *intrinsic geometric structures* in the data set, and taking powers of the matrix P is equivalent to integrating the local geometry of the data at different scales.

An equivalent way to look at powers of P is to make use of its eigenvectors and eigenvalues: it can be showed that there exists a sequence $1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq ...$ of eigenvalues and a collection $\{\psi_0, \psi_1, \psi_2, ...\}$ of (right) eigenvectors for P:

$$P\psi_l = \lambda_l \psi_l \, .$$

These eigenvalues and eigenvectors provide embedding coordinates for the set Ω. The data points can be mapped into a Euclidean space via the embedding

$$\Psi_t : x \longmapsto \left\langle \lambda_1^t \psi_1(x), \ldots, \lambda_{m(t)}^t \psi_{m(t)}(x) \right\rangle \ , \tag{2.1}$$

where $t \geq 0$. Discussions regarding the number $m(t)$ of diffusion coordinates to employ and concerning the connection with the so-called diffusion distance are provided in [22, 26, **?**].

Next, we address the issue of obtaining a density-invariant embedding. The point is to make the embedding reflect only the geometry of the data and be insensitive to the distribution of the points. Classical eigenmap methods [5, 6, 7, 27]**,** provide an embedding that combines the information of both the density and geometry, and the embedding coordinates heavily depend on the density of the data points. In order to remove the influence of the distribution of the data points, we renormalize the Gaussian edge weights $w_\varepsilon(\cdot,\cdot)$ with an estimate of the density. This is summarized in Algorithm 1 which was first introduced and analyzed in [22].

Algorithm 1 Approximation of the Laplace-Beltrami diffusion

```
Start with a rotation-invariant kernel $w_\varepsilon(x_i, x_j) = h\left(\frac{\|x_i - x_j\|^2}{\varepsilon}\right)$.
Let
    $q_\varepsilon(x_i) \triangleq \sum_{x_j \in \Omega} w_\varepsilon(x_i, x_j)$ ,
    and form the new kernel
    $\widetilde{w}_\varepsilon(x_i, x_j) = \frac{w_\varepsilon(x_i, x_j)}{q_\varepsilon(x_i) q_\varepsilon(x_j)}$ .        (2.2)
Apply the normalized graph Laplacian construction to this kernel, i.e., set
    $d_\varepsilon(x) = \sum_{z \in \Omega} \widetilde{w}_\varepsilon(x_i, x_j)$ ,
    and define the anisotropic transition kernel
    $p_\varepsilon(x_i, x_j) = \frac{\widetilde{w}_\varepsilon(x_i, x_j)}{d_\varepsilon(x_i)}$ .
```

Next we describe the data fusion scheme, where, for the sake of clarity, we direct our discussion to the case of two input channels, while it can be easily extended to an arbitrary number of them. Suppose one has two sets of measurements related to a particular phenomenon $\Omega = \{x_1, ..., x_n\}$. Denote these sets of measurements $\Omega_1 = \{y_1^1, ..., y_n^1\}$ and $\Omega_2 = \{y_1^2, ..., y_n^2\}$, respectively, where y_i^1 and y_i^2 are high-dimensional measurements. We aim to fuse Ω_1 and Ω_2 by computing a unified low-dimensional representation $\widehat{\Omega} = \{z_1, ..., z_n\}$. Note that we assume that $\Omega_1$and Ω_2 are aligned, meaning that y_i^1 and y_i^2 relate to the same instance x_i. When this assumption is invalid, one has to apply a multi-sensor alignment scheme [12] prior to applying the fusion procedure.

We start by computing the Laplace-Beltrami embeddings of Ω_1 and Ω_2 denoted $\Phi_1^{m_1} = \{\phi_1^1, ..., \phi_n^1)\}$ and $\Phi_2^{m_2} = \{\phi_1^2, ..., \phi_n^2)\}$, respectively, where m_i is the dimensionality of each embedding. Each representation reflects the geometry of the data as viewed by each sensor individually. In order to combine these analyzes into a unified representation $\widehat{\Omega}$, we form $\widehat{\Omega} = \{z_1, ..., z_n\}$ where

$$z_i = \{\phi_i^1, \phi_i^2\}, \tag{2.3}$$

ϕ_i^1 and ϕ_i^2 being of dimensions m_1 and m_2, respectively. In general, given K input sources we have

$$z_i = \{\phi_i^1, \ldots, \phi_i^K\}. \tag{2.4}$$

This boils down to combining the embedding coordinates corresponding to each sample x_i over the different input channels $\{\Omega_i\}$.

In essence, our scheme is the embedding analogue of boosting [28], where instead of adaptively integrating the output of several classifiers, we combine different embeddings. In particular, one can consider an equivalent to the *AdaBoost* scheme [28] for semi-supervised classification, where Eq. 2.4 is replaced with

$$\widehat{z_i} = \{a^1\phi_i^1, \ldots, a^K\phi_i^K\}, \tag{2.5}$$

$\left\{a^1, \ldots, a^K\right\}$ being the weights per embedding. In that sense, the embeddings z_i can be considered as different features, and one can apply a standard implementation of *AdaBoost* to Eq. 2.4. Yet, in this work, the focal point is to derive general-purpose coordinates regardless of a particular application. The scheme is summarized in Algorithm 2.

Algorithm 2 Multisensor embedding

```
Starting with $K$ input sources $\Omega_k = \{y_1^k, ..., y_n^k\}$, $k = 1...K$.
Compute the Laplace-Beltrami embeddings of $\{\Omega_k\}$, denoted $\Phi_k^{m_k}$, where $m_k$ is the dimensionality of the embedding of the $k$'th channel.
Compute the unified coordinates set $\widehat{\Omega} = \{z_1, ..., z_n\}$ by appending the embeddings of each input sensor
    $$z_i = \{\phi_i^1, \ldots, \phi_i^K\}, i = 1...n, \; 2\, k = 1...K.$$
```

3 Experimental results

The proposed scheme was experimentally verified by applying it to two tasks. First, we extend our former results in visual-only lip-reading [4] to audio-visual data. The audio-visual inputs are integrated using the multisensor fusion scheme given in Section 2 and used for spoken-digit recognition. We show that the fused multi-sensor representation provides better recognition. Second, we integrate several image cues (texture, RGB values, contours etc.) and show that using them in conjugation improves the segmentation results.

3.1 Spoken-digit recognition

We start by providing a short description of the experimental setup. We follow the statistical learning scheme used in [4], where the classifier was constructed in two steps. First we parametrized the embedding manifold using a large number of unlabeled samples. The embedding is then extended, using the Geometric Harmonics [4, 29], to a small set of labeled examples to create a set of "signatures" in the embedding coordinates. Then, given a test sample, we embed it by extending the manifold embedding, and find the nearest signature in the embedding space.

To this end, we recorded several grayscale movies depicting the lips of a subject reading a text in English and retained both the video sequence and the audio track. Each video frame was cropped into a rectangular of size 140×110 around the lips and was viewed as a point in $\mathbb{R}^{140\times 110}$. As far as the audio data was concerned, the sound signal was broken up into overlapping time-windows centered at the beginning of each video frame. The sampling rate of the video begin 25 frames per seconds, we chose to form windows of duration equal to 8 ms, that is, equal to the duration of two video frames. In order to reduce the influence of this splitting, each piece of signal was multiplied by a bell-shaped function and we then computed the DCT of the result. Last, we considered the logarithm of the magnitude of this function as being the audio features. The audio and video data sets therefore contained the same number of points.

The first data set consisted of 6000 video frames (and as many audio windows), corresponding to the speaker reading a press article. We will refer to this data as "text data". Next, we asked the subject to repeat each digit "zero", "one", ... , "nine" 40 times. This was used to construct a small vocabulary of words later employed for training and testing a simple classifier. To each spoken digit corresponded a sequence of frames in the video data, and a sequence of time-windows for the audio data. We will refer to this data as "digit data".

We proceeded as follows for each channel: first, the data points corresponding to the text data were used to learn the geometry of speech data as we formed a graph with Gaussian weights $\exp(\frac{\|x_i - x_j\|^2}{\varepsilon})$ on the edges, for an appropriately chosen scale $\varepsilon > 0$. We then renormalized the Gaussian weights using the Laplace-Beltrami normalization described in Algorithm 1. In order to obtain a low-dimensional parametrization we computed the diffusion coordinates on this new graph. Therefore we ended up with two embeddings, corresponding to either the audio or visual data.

The next step involved the digits data. We computed the diffusion coordinates for all of the samples in the digits data, by applying the Geometric Harmonics scheme [4, 29] and extending the diffusion coordinates computed on the text data.

In order to train a classifier for digit identification, we randomly selected 20 sequences of each digit, the remaining sequences being used as a test set. Each digit word can now be viewed as a trajectory in the diffusion space and the word recognition problem now amounts to identifying trajectories in the diffusion space (see Fig. 1). We can now build a classifier based on comparing a new trajectory to a collection of labeled trajectories in the training set. In order to compare trajectories in the diffusion space we used the symmetric Hausdorff distance between two sets Γ_1 and Γ_2, defined as

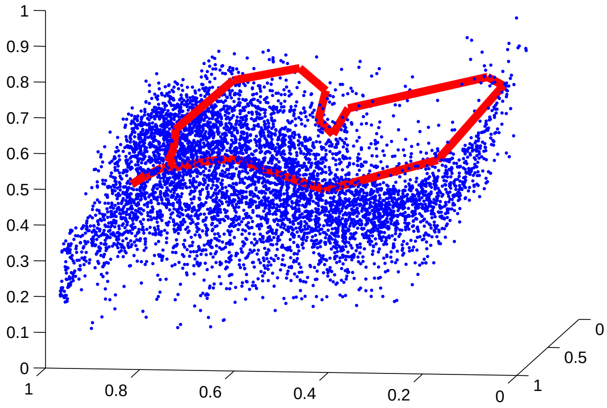


Figure 1: The visual data in the first 3 diffusion coordinates. We also represented a trajectory corresponding to an instance of the word “one”.

$$d_H(\Gamma_1, \Gamma_2) = \max \left\{ \max_{x_2 \in \Gamma_2} \min_{x_1 \in \Gamma_1} \{\|x_1 - x_2\|\}, \max_{x_1 \in \Gamma_1} \min_{x_2 \in \Gamma_2} \{\|x_1 - x_2\|\} \right\} . \tag{3.1}$$

Results of this classifier for the visual data only were already presented in [4], where 15 eigenvectors were used for the embedding. For the sake of completeness, we re-ran this experiment with

10 eigenvectors. The results are shown in Table 1. Concerning the audio data, the classification performance table for a classifier using 10 eigenvectors is presented in Table 2.

	"0"	**"1"**	**"2"**	**"3"**	**"4"**	**"5"**	**"6"**	**"7"**	**"8"**	**"9"**
zero	**0.90**	0	0	0.01	0	0	0.08	0	0	0
one	0	**0.99**	0	0	0	0	0	0	0.01	0
two	0.04	0.01	**0.90**	0.03	0.02	0	0	0	0	0
three	0	0	0.01	**0.94**	0	0	0.01	0.02	0.01	0
four	0.01	0	0	0.05	**0.93**	0	0	0	0	0
five	0	0	0	0	0	**0.81**	0.01	0.16	0	0.01
six	0.07	0	0	0.01	0	0	**0.87**	0.03	0.01	0.01
seven	0.03	0	0	0.04	0	0.07	0.05	**0.74**	0.04	0.02
eight	0	0	0	0	0.02	0.03	0	0.03	**0.75**	0.16
nine	0	0	0	0	0	0	0	0.04	0.14	**0.82**

Table 1: Visual only based classifier performance, averaged over 50 random trials and using 10 diffusion coordinates. Each row corresponds the classification distribution of a given digit over then 10 classes. The data set was embedded in 15 dimensions.

Finally, in order to illustrate the superiority of combining both data channels using our multi-sensor integration scheme, we present the results obtained when using Algorithm 2 (see Table 3). More precisely, we appended the first 5 eigenvectors of the audio data with the top 5 eigenvectors of the video data, and then computed a new graph from this new feature representation of the data. Finally a classifier was trained and tested on an embedding using 10 eigenvectors of the diffusion defined on this new graph (see Fig. **??**).

A summary of all performances is also shown in Table 4. Clearly, our scheme combining both channels outperforms the classifiers using only one channel. More precisely, it seems to

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.75**	0	0.04	0	0.01	0.01	0.06	0.08	0.05	0
one	0	**0.94**	0	0	0	0.03	0	0	0	0.02
two	0.02	0	**0.87**	0.04	0.01	0	0.01	0	0.03	0.02
three	0.01	0	0.03	**0.90**	0.02	0.01	0	0	0.01	0.01
four	0.01	0	0	0.02	**0.96**	0	0	0	0	0.01
five	0.01	0.01	0	0.06	0	**0.86**	0	0.01	0.01	0.03
six	0	0	0	0	0.01	0	**0.93**	0.05	0	0
seven	0.05	0	0	0	0	0	0.14	**0.81**	0.01	0
eight	0.02	0	0.04	0.02	0	0.02	0	0.07	**0.80**	0.03
nine	0	0.01	0	0.01	0.01	0.04	0	0	0.01	**0.92**

Table 2: Audio only based classifier performance, averaged over 50 random trials and using 10 diffusion coordinates. Each row corresponds the classification distribution of a given digit over then 10 classes. The data set was embedded in 15 dimensions.

get the best of the predictive powers of the audio and visual classifiers. In fact, this is a straight consequence of the concatenation of the audio and visual diffusion features. For instance, “one” is very successfully classified using the visual channel. As suggested in [4], typical frame sequences corresponding to the word “one” contain pictures with an open mouth and no teeth appearing **STEPHANE: ADD A FEW PICTURES ILLUSTRATING THIS POINT**. This type of frame almost never appear in other digit sequences. As a consequence, trajectories for the word “one” will be well separated from other digit trajectories in the visual diffusion space **STEPHANE: SHOW TRAJECTORY PIC**. As far as audio is concerned, the separation is not so important and there is some amount of confusion with “five” and “nine”. When appending both the audio and visual representations, the separation remains high.

Notice also that these good results were obtained despite the fact that we used only 5 eigenvectors from each channel in the combined scheme, when 10 eigenvectors were used for either of the single-channel schemes.

3.2 Image segmentation

The sensor fusion scheme was also applied to multi-cue image segmentation. As features we used combinations of Interleaving Contours (IC) [30], the L_2 metric between RGB values and Gabor filters based texture descriptors [31]. The Gabor filters used 3 scales and 8 orientations. For each pixel, the metrics were computed in an area of 5×5 pixels around P. Such that given a feature $f(i,j)$ computed over the image I, the distance between the pixels $P\left(i,j\right)$ and $P_1\left(i_1,j_1\right)$ with

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.90**	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.00	0.00
one	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
two	0.00	0.00	**0.96**	0.01	0.02	0.00	0.00	0.00	0.00	0.00
three	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
four	0.00	0.00	0.00	0.04	**0.96**	0.00	0.00	0.00	0.00	0.00
five	0.00	0.00	0.00	0.00	0.00	**0.97**	0.00	0.00	0.02	0.01
six	0.06	0.00	0.00	0.00	0.00	0.00	**0.90**	0.04	0.00	0.00
seven	0.03	0.00	0.00	0.00	0.00	0.00	0.03	**0.93**	0.00	0.00
eight	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	**0.95**	0.03
nine	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.02	**0.96**

Table 3: **Combined**: Classification results for the scheme combining both channels, over 50 random trials. The combined graph was built from a feature representation of the data based on appending the first 5 eigenvectors of the audio channel with the first 5 eigenvectors of the video stream. From this graph, we computed 10 eigenvectors, and we used them for representing the data.

Channel type	**“0”**	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
Audio	0.75	0.94	0.87	0.90	**0.96**	0.86	**0.93**	0.81	0.80	0.92
Visual	**0.90**	**0.99**	0.90	0.94	0.93	0.81	0.87	0.74	0.75	0.82
Combined	**0.90**	**0.99**	**0.96**	**0.99**	**0.96**	**0.97**	0.90	**0.93**	**0.95**	**0.96**

Table 4: Summary

respect to the feature is given by

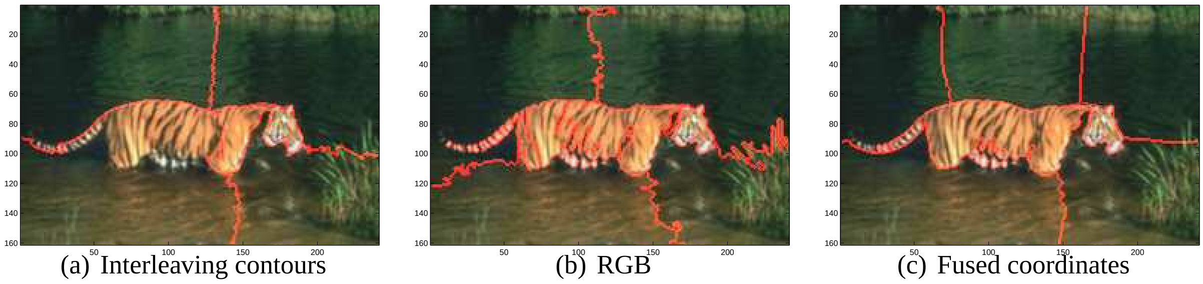
$$D\left(P,P_1\right)=\begin{cases}\left\|f(i,j)-f(i_1,j_1)\right\|_{L_2} & \sqrt{(i-i_1)^2+(j-j_1)^2}\leq 2\\ \infty & else\end{cases}. \tag{3.2}$$

Equation 3.2 is used to sparsify the affinity matrix of I, otherwise, its eigenvectors computation become computationally exhaustive for common image sizes. Applying Eq. 3.2 might create spurious additional parameterizations related to the spatial coordinates. For instance, consider the vertical and horizontal lines in all of the segmentations in Fig. 2. We refrained from using the Nyström Method [32], that would have resolved this issue, in order to simplify the testing procedure and as this phenomenon is well understood.

For every input image, we computed several embeddings and the integrated representation was computed by the procedure described in Section 2. We emphasize, that for each image, the same embedding vectors were used both for the single and multi-cue segmentations. In all of the simulations we used 5 eigenvectors from each feature. For all images we present the segmentation results of applying k-means clustering to each of the original embeddings and the fused coordinates. This follows the Modified-NCut (MNCut) image segmentation scheme [33]. The scheme was implemented in Matlab and used the built-in kmeans and SVD implementations. Note that the regular Graph-Laplacian was used for the segmentation and not the density-invariant Laplace-Beltrami.

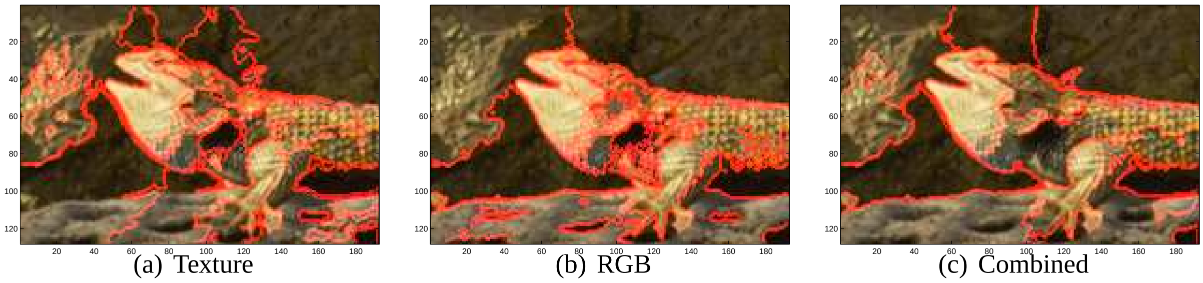
Figure 2 depicts the segmentation results of the *Tiger* image taken from the Berkeley segmentation database. The images were segmented using the IC and RGB features and the result are shown Figs. 2a and 2b, respectively. The segmentation results in Fig. 2c show that using the using fused coordinates provided better results than either the IC or the RGB segmentation results.

Different features were used in Fig. 3. The IC feature is inefficient in analyzing highly-textured images, as it creates over-segmentation. Thus, we used the RGB and texture features. The texture based segmentation (Fig. 3a) results in over-segmentation in the lizard's body, while missing the cut between the front and background rocks on the left side of the image. Similarly, using the RGB



(a) Interleaving contours (b) RGB (c) Fused coordinates

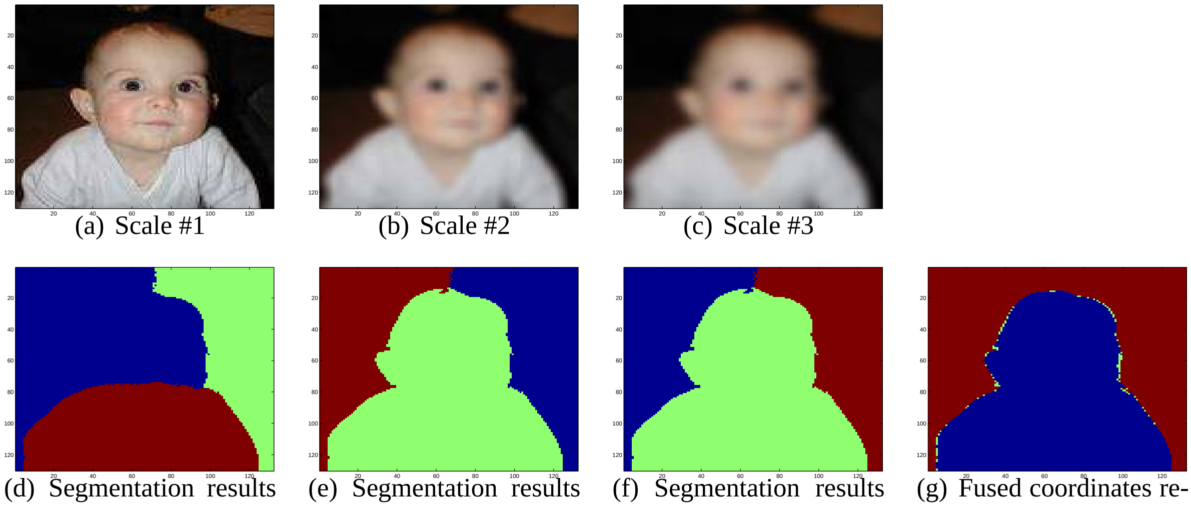
Figure 2: Applying the proposed scheme to the *Tiger* image. (a) Segmentation achieved using the Interleaving contours edge based features. (b) Segmentation results based on L_2 differences in RGB values. (c) Using the fused coordinates we achieve a visually better pleasing result.



(a) Texture (b) RGB (c) Combined

Figure 3: Applying the proposed scheme to the *Lizard* image. (a) Segmentation achieved using the texture features. Note the over segmentation in the are behind the Lizard's head. (b) Segmentation results based on L_2 differences in RGB values. Note the over-segmentation above the Lizard's leg. (c) Using the fused coordinates we achieve a visually better pleasing result.

descriptor also results in over-segmentation. In contrast, the combined segmentation is better eye pleasing and is able to detect salient multi-cue edges in the image.



(a) Scale #1

(b) Scale #2

(c) Scale #3

(d) Segmentation results at Scale #1

(e) Segmentation results at Scale #2

(f) Segmentation results at Scale #3

(g) Fused coordinates results

Figure 4: Multisensor embedding applied to multiscale image segmentation. The Interleaving contours edge based feature was applied to each of the image in the first row ((a),(b),(c)). The second row depicts the corresponding segmentation results. (g) show the improved segmentation achieved using the fused coordinates.

Finally, we applied the fusion scheme to multi-scale image segmentation. The image was smoothed by a Gaussian kernel and three resolution scales (shown in Figs. 4a, 4b and 4c) were created. The IC feature was computed based on each image and the embeddings were fused. We see that using the proposed scheme resulted in a segmentation that combined the mutual cluster boundaries in the image, allowing to overlook some of the spurious segmentations, such as the left eye in Fig. 4a and the throat area in 4b. In [17] the multiscale segmentation was computed via a computation of an “average cut”. There, the Markov matrices that were computed at each scale were used, rather than the embedding vectors. In practice, there is no difference between the multiscale fusion and the fusion of the other descriptors depicted in Figs. 2 and 3. In particular one can combine different features and scales directly.

To conclude, by fusing the different features, we were able to achieve better segmentation results. In essence this approach resembles the biological vision systems by combining different cues and emphasizing salient multi-features edges. The scheme is flexible and once the embeddings of each feature are computed, one can combine the embeddings in any possible way without having to recompute them.

4 Conclusions and future work

In this work we presented a unified multisensor data embedding scheme, based on the diffusion framework. The fusion was achieved by combining the embeddings of different input channels. We applied the scheme to audio-visual lip reading and image segmentation that are typical examples of multisensor pattern recognition and classification. In both cases, the results achieved by using fused coordinates were superior to those of the single sensor.

We embedded each data source separately and then appended the embeddings to produce the fused representation. Although this approach is straightforward and allows to combine different channels easily, it is possible that different channels are correlated. Then, one can find a lower dimensional representation by considering the unified coordinates as a the features of a signal and re-embedding them to further reduce the dimensionality.

The image segmentation results, suggest that in certain applications, one can utilize a variety of features in different resolution scales. Thus, due to the large number of possible input channels, it might be beneficial to compute adaptive weights that maximize a certain criterion. For instance, in semi-supervised classification problems, one can train the weights of the combined representation for optimal classification over a training set by using the *AdaBoost* algorithm.

References

[1] E. Kidron, Y. Schechner, and M. Elad, "Pixels that sound," in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, vol. I, June 2005, pp. pp. 88–96.

[2] J. Driver, "Enhancement of selective listening by illusory mislocation of speech sounds due to lip-reading," *Nature*, no. 381, pp. 66–68, 1996.

[3] Y. Gutfreund, W. Zheng, and E. I. Knudsen, "Gated visual input to the central auditory system," *Science*, no. 297, pp. 1556–1559, 2002.

[4] S. Lafon, Y. Keller, A. Glaser, and R. R. Coifman, "Data fusion and multi-cue data matching by diffusion maps," *Submitted for publication*.

[5] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[6] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 6, no. 15, pp. 1373–1396, June 2003.

[7] D. Donoho and C. Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data," *Proceedings of the National Academy of Sciences*, vol. 100, no. 10, pp. 5591–5596, May 2003.

[8] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignement," Department of computer science and engineering, Pennsylvania State University, Tech. Rep. CSE-02-019, 2002.

[9] H. Li, B. S. Manjunath, and S. K. Mitra, "A contour-based approach to multisensor image registration," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 320–334, March 1995.

[10] R. Sharma and M. Pavel, “Registration of video sequences from multiple sensors,” in *Proceedings of the Image Registration Workshop*. NASA GSFC, 1997, pp. 361–366.

[11] A. Gueziec, X. Pennec, and N. Ayache, “Medical image registration using geometric hashing,” *IEEE Computational Science & Engineering, special issue on Geometric Hashing*, vol. 4, no. 4, pp. 29–41, October-December 1997.

[12] J. Ham, D. Lee, and L. Saul, “Semisupervised alignment of manifolds,” in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados*, pp. 120–127.

[13] E. Waltz and J. Llinas, *Spectral graph theory*. Artech House, Boston, 1990.

[14] J. Sasiadek, “Sensor fusion,” *Annual reviews in control*, vol. 26, pp. 203–228, 2002.

[15] A. Fred and A. K. Jain, “Combining multiple clusterings using evidence accumulation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 835–850, June 2005.

[16] Y. Weiss, “Segmentation using eigenvectors: A unifying view,” in *ICCV ’99: Proceedings of the International Conference on Computer Vision*, vol. 2. Washington, DC, USA: IEEE Computer Society, 1999, p. 975.

[17] S. X. Yu, “Segmentation using multiscale cues.” in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), 27 June - 2 July 2004, Washington, DC, USA*, 2004, pp. 247–254.

[18] T. Cour, F. Bénézit, and J. Shi, “Spectral segmentation with multiscale graph decomposition.” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, 2005, pp. 1124–1131.

[19] S. X. Yu, “Segmentation induced by scale invariance,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, vol. 1, 2005, pp. 444 – 451.

[20] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, “Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7426–7431, May 2005.

[21] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, “Geometric diffusions as a tool for harmonics analysis and structure definition of data: Multiscale methods,” *Proceedings of the National Academy of Sciences*, vol. 102, no. 21, pp. 7432–7437, May 2005.

[22] R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[23] F. Chung, *Spectral graph theory*. CBMS-AMS, May 1997, no. 92.

[24] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: a geometric framework for learning from examples,” University of Chicago, Tech. Rep. TR-2004-06, 2004.

[25] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Tran PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[26] B. Nadler, S. Lafon, R. Coifman, and I. Kevrekidis, “Diffusion maps, spectral clustering and the reaction coordinates of dynamical systems,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[27] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, “The Isomap algorithm and topological stability,” *Science*, vol. 295, no. 5552, p. 7, 2002.

[28] T. Hastie, R. Ribshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer.

[29] R. Coifman and S. Lafon, “Geometric harmonics,” *Applied and Computational Harmonic Analysis*, 2005, to appear.

[30] M. Meila and J. Shi, “Learning segmentation by random walks.” in *NIPS*, 2000, pp. 873–879.

[31] J. Malik, S. Belongie, T. Leung, and J. Shi, “Contour and texture analysis for image segmentation,” *International Journal of Computer Vision*, vol. 43, no. 1, pp. 7–27, 2001.

[32] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, “Spectral grouping using the nyström method.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.

[33] M. Maila and J. Shi, “A random walks view of spectral segmentation,” in *AI and STATISTICS (AISTATS) 2001*, 2001.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

GEOMETRIC DIFFUSIONS AS A TOOL FOR HARMONIC ANALYSIS AND STRUCTURE DEFINITION OF DATA

PART I: DIFFUSION MAPS

R. R. COIFMAN, S. LAFON, A. B. LEE, M. MAGGIONI, B. NADLER, F. WARNER, AND S.W. ZUCKER

Abstract. We provide a framework for structural multiscale geometric organization of graphs and subsets of $\mathbb{R}^n$. We use diffusion semigroups to generate multiscale geometries in order to organize and represent complex structures. We show that appropriately selected eigenfunctions or scaling functions of Markov matrices, which describe local transitions, lead to macroscopic descriptions at different scales. The process of iterating or diffusing the Markov matrix is seen as a generalization of some aspects of the Newtonian paradigm, in which local infinitesimal transitions of a system lead to global macroscopic descriptions by integration. In Part I below, we provide a unified view of ideas from data analysis, machine learning and numerical analysis. In the second part of this paper, we augment this approach by introducing fast order-N algorithms for homogenization of heterogeneous structures as well as for data representation.

1. Introduction

The geometric organization of graphs and data sets in $\mathbb{R}^n$ is a central problem in statistical data analysis. In the continuous Euclidean setting, tools from harmonic analysis, such as Fourier decompositions, wavelets and spectral analysis of pseudo-differential operators have proven highly successful in many areas such as compression, denoising and density estimation [1, 2]. In this paper, we extend multiscale harmonic analysis to discrete graphs and subsets of $\mathbb{R}^n$. We use diffusion semigroups to define and generate multiscale geometries of complex structures. This framework generalizes some aspects of the Newtonian paradigm, in which local infinitesimal transitions of a system lead to global macroscopic descriptions by integration — the global functions being characterized by differential equations. We show that appropriately selected eigenfunctions of Markov matrices (describing local transitions, or affinities in the system) lead to macroscopic representations at different scales. In particular, the top eigenfunctions permit a low-dimensional geometric embedding of the set into $\mathbb{R}^k$, with $k \ll n$, so that the ordinary Euclidean distance in the embedding space measures intrinsic diffusion metrics on the data. Many of these ideas appear in a variety of contexts of data analysis, such as spectral graph theory, manifold learning, nonlinear principal components and kernel methods. We augment these approaches by showing that the diffusion distance is a key intrinsic geometric quantity linking spectral theory of the Markov process, Laplace operators, or kernels, to the corresponding geometry and density of the data. This opens the door to the application of methods from numerical analysis and signal processing to the analysis of functions and transformations of the data.

2. DIFFUSIONS MAPS

The problem of finding meaningful structures and geometric descriptions of a data set X is often tied to that of dimensionality reduction. Among the different techniques developed, particular attention has been paid to kernel methods [3]. Their nonlinearity as well as their locality-preserving property are generally viewed as a major advantage over classical methods like Principal Component Analysis and classical Multidimensional Scaling. Several other methods to achieve dimensional reduction have also emerged from the field of manifold learning, e.g. Local Linear Embedding [4], Laplacian eigenmaps [5], Hessian eigenmaps [6], Local Tangent Space Alignment [7]. All these techniques minimize a quadratic distortion measure of the desired coordinates on the data, naturally leading to the eigenfunctions of Laplace type operators as minimizers. We extend the scope of application of these ideas to various tasks, such as regression of empirical functions, by adjusting the infinitesimal descriptions, and the description of the long-time asymptotics of stochastic dynamical systems.

The simplest way to introduce our approach is to consider a set X of normalized data points. Define the "quantized" correlation matrix $C = \{c_{ij}\}$, where $c_{ij} = 1$ if $(x_i \cdot x_j) > 0.95$, and $c_{ij} = 0$ otherwise. We view this matrix as the adjacency matrix of a graph on which we define an appropriate Markov process to start our analysis. A more continuous kernel version can be defined as $c_{ij} = e^{\frac{1-(x_i \cdot x_j)}{\varepsilon}} = e^{-\frac{\|x_i - x_j\|^2}{2\varepsilon}}$. The remarkable fact is that the eigenvectors of this "corrected correlation" can be much more meaningful in the analysis of data than the usual principal components as they relate to diffusion and inference on the data.

As an illustration of the geometric approach, suppose that the data points are *uniformly* distributed on a manifold X. Then it is known from spectral graph theory [8] that if $W = \{w_{ij}\}$ is any symmetric positive semi-definite matrix, with non-negative entries, then the minimization of

$$Q(f) = \sum_{i,j} w_{ij}(f_i - f_j)^2,$$

where f is a function on the data set X with the additional constraint of unit norm, is equivalent to finding the eigenvectors of $D^{-\frac{1}{2}}WD^{\frac{1}{2}}$, where $D = \{d_{ij}\}$ is a diagonal matrix with diagonal entry d_{ii} equal to the sum of the elements of W along the i^{th} row. Belkin *et al* [5] suggest the choice $w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\varepsilon}}$, in which case the distortion Q clearly penalizes pairs of points that are very close, forcing them to be mapped to very close values by f. Likewise, pairs of points that are far away from each other play no role in this minimization. The first few eigenfunctions $\{\phi_k\}$ are then used to map the data in a nonlinear way so that the closeness of points is preserved. We will provide a principled geometric approach for the selection of eigenfunction coordinates.

This general framework based upon diffusion processes leads to efficient multiscale analysis of data sets for which we have a Heisenberg localization principle relating localization in data to localization in spectrum. We also show that spectral properties can be employed to embed the data into a Euclidean space via a *diffusion map*. In this space, the data points are reorganized in such a way that the Euclidean distance corresponds to a *diffusion metric*. The case of submanifolds of $\mathbb{R}^n$ is studied in greater detail and we show how to define different kinds of diffusions in order to recover the intrinsic geometric structure, separating geometry from statistics. More details on the topics covered in this section can be found in [9]. We also propose an additional diffusion map based on a specific anisotropic kernel whose eigenfunctions capture the long-time asymptotics of data sampled from a stochastic dynamical system [10].

2.1. **Construction of the diffusion map.** From the above discussion, the data points can be thought of as being the nodes of a graph whose weight function $k(x, y)$ (also referred to as "kernel" or "affinity function") satisfies the following properties:

- k is symmetric: $k(x, y) = k(y, x)$,
- k is positivity preserving: for all x and y in X, $k(x, y) \geq 0$,
- k is positive semi-definite: for all real-valued bounded functions f defined on X,
$$\int_X \int_X k(x, y) f(x) f(y) d\mu(x) d\mu(y) \geq 0\,,$$
where μ is a probability measure on X.

The construction of a diffusion process on the graph is a classical topic in spectral graph theory (weighted graph Laplacian normalization, see [8]), and the procedure consists in renormalizing the kernel $k(x, y)$ as follows: for all $x \in X$,

$$\text{let} \quad v(x) = \int_X k(x, y) d\mu(y)\,,$$

and set

$$a(x, y) = \frac{k(x, y)}{v(x)}\,.$$

Notice that we have the following conservation property:

$$\int_X a(x, y) d\mu(y) = 1\,, \tag{2.1}$$

therefore, the quantity $a(x, y)$ can be viewed as the probability for a random walker on X to make a step from x to y. Now we naturally define the diffusion operator

$$Af(x) = \int_X a(x, y) f(y) d\mu(y)\,.$$

As is well known in spectral graph theory [8], there is a spectral theory for this Markov chain, and if $\widetilde{A}$ is the integral operator defined on $L^2(X)$ with the kernel

$$\widetilde{a}(x, y) = a(x, y)\sqrt{\frac{v(x)}{v(y)}} \tag{2.2}$$

then it can be verified that $\widetilde{A}$ is a symmetric operator. Consequently, we have the following spectral decomposition

$$\widetilde{a}(x, y) = \sum_{i \geq 0} \lambda_i^2 \phi_i(x) \phi_i(y)\,, \tag{2.3}$$

where $\lambda_0 = 1 \geq \lambda_1 \geq \lambda_2 \geq ....$ Let $\widetilde{a}^{(m)}(x, y)$ be the kernel of $\widetilde{A}^m$. Then we have

$$\widetilde{a}^{(m)}(x, y) = \sum_{i \geq 0} \lambda_i^{2m} \phi_i(x) \phi_i(y)\,. \tag{2.4}$$

Last we introduce the family of *diffusion maps* $\{\Phi_m\}$ by

$$\Phi_m(x) = \begin{pmatrix} \lambda_0^m \phi_0(x) \\ \lambda_1^m \phi_1(x) \\ \vdots \end{pmatrix}\,,$$

and the family of *diffusion distances* $\{D_m\}$ defined by

$$D_m^2(x, y) = \widetilde{a}^{(m)}(x, x) + \widetilde{a}^{(m)}(y, y) - 2\widetilde{a}^{(m)}(x, y)\,.$$

The quantity $a(x, y)$, which is related to $\widetilde{a}(x, y)$ according to equation (2.2), can be interpreted as the transition probability of a diffusion process, while $a^{(m)}(x, y)$ represents the probability of transition from x to y in m steps. To this diffusion process corresponds the distance $D_m(x, y)$ which defines a metric on the data that measures the rate of connectivity of the points x and y by paths of length m in the data, and in particular, it is small if there are a large number of paths connecting x and y. Note that, unlike the geodesic distance, this metric is robust to perturbations on the data.

The dual point of view is that of the analysis of functions defined on the data. The kernel $\widetilde{a}^{(m)}(x, \cdot)$ can be viewed as a bump function centered at x, that becomes wider as m increases. The distance $D_{2m}(x, y)$ is also a distance between the two bumps $\widetilde{a}^{(m)}(x, \cdot)$ and $\widetilde{a}^{(m)}(y, \cdot)$:

$$D_{2m}^2(x, y) = \int_X |\widetilde{a}^{(m)}(x, z) - \widetilde{a}^{(m)}(y, z)|^2 dz\,.$$

The eigenfunctions have the classical interpretation of an orthonormal basis, and their frequency content can be related to the spectrum of operator A in what constitutes a *generalized Heisenberg principle*. The key observation is that, for many practical examples, the numerical rank of the operator $A^{(m)}$ decays rapidly as seen from equation (2.4) or from Figure 1. More precisely, since $0 \leq \lambda_i \leq \lambda_0 = 1$, the kernel $\widetilde{a}^{(m)}(x, y)$, and therefore the distance $D_m(x, y)$, can be computed to high accuracy with only a few terms in the sum of (2.4), that is to say, by only retaining the eigenfunctions ϕ_i for which λ_i^{2m} exceeds a certain precision threshold. Therefore, the rows (the so-called bumps) of A^m span a space of lower numerical dimension, and the set of columns can be downsampled. Furthermore, to generate this space, one just needs the top eigenfunctions, as prescribed in equation (2.4). Consequently, by a change of basis, eigenfunctions corresponding to eigenvalues at the beginning of the spectrum have low frequencies, and the number of oscillations increase as one moves further down in the spectrum.

The link between diffusion maps and distances can be summarized by the spectral identity

$$\|\Phi_m(x) - \Phi_m(y)\|^2 = \sum_{j \geq 0} \lambda_j^{2m}(\phi_j(x) - \phi_j(y))^2 = D_m^2(x, y)\,,$$

which means that the diffusion map Φ_m embeds the data into a Euclidean space in which the Euclidean distance is equal to the diffusion distance D_m. Moreover, the diffusion distance can be accurately approximated by retaining only the terms for which λ_j^{2m} remains numerically significant: the embedding

$$x \longmapsto \breve{x} = (\lambda_0^m \phi_0(x), \lambda_1^m \phi_1(x), ..., \lambda_{j_0}^m \phi_{j_0}(x))$$

satisfies

$$\begin{aligned} D_m^2(x, y) &= \sum_{j=0}^{j_0 - 1} \lambda_j^{2m}(\phi_j(x) - \phi_j(y))^2 \left(1 + \mathcal{O}(e^{-\alpha m})\right) \\ &= \|\breve{x} - \breve{y}\|^2 (1 + \mathcal{O}(e^{-\alpha m}))\,. \end{aligned}$$

Therefore there exists an m_0 such that for all $m \geq m_0$, the diffusion map with the first j_0 eigenfunctions embeds the data into $\mathbb{R}^{j_0}$ in an approximately isometric fashion, with respect to the diffusion distance D_m.

2.2. **The heat diffusion map on Riemannian submanifolds.** Suppose that the data set X is approximately lying along a submanifold $\mathcal{M} \subset \mathbb{R}^n$, with a density $p(x)$ (not necessarily uniform on $\mathcal{M}$). This kind of situation arises in many applications ranging from hyperspectral imagery to image processing to vision. For instance, in the latter field, a model for edges can be generated by considering pixel neighborhoods whose variability is governed by a few parameters [11, 12].

We consider isotropic kernels, *i.e.*, kernels of the form

$$k_\varepsilon(x, y) = h\left(\frac{\|x - y\|^2}{\varepsilon}\right).$$

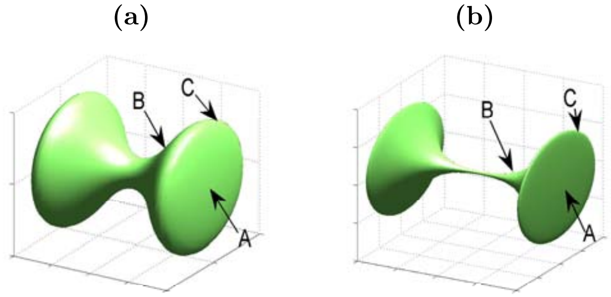


FIGURE 2. A dumbbell (a) is embedded using the first 3 eigenfunctions (b). Because of the bottleneck, the two lobes are pushed away from each other. Observe also that in the embedding space, point A is closer to the handle (point B) than any point on the edge (like point C), as there are many more short paths joining A and B than A and C.

In [5], Belkin *et al* suggest to take $k_\varepsilon(x, y) = e^{-\frac{\|x-y\|^2}{\varepsilon}}$ and to apply the weighted graph Laplacian normalization procedure described in the previous section. They show that if the density of points is uniform, then as $\varepsilon \to 0$, one is able to approximate the Laplace-Beltrami operator Δ on $\mathcal{M}$.

However when the density p is not uniform, as is often the case, the limit operator is conjugate to an elliptic Schrödinger-type operator having the more general form $\Delta + Q$, where $Q(x) = \frac{\Delta p(x)}{p(x)}$ is a potential term capturing the influence of the non-uniform density. By writing the non-uniform density in a Boltzmann form, $p(x) = e^{-U(x)}$, the infinitesimal operator can be expressed as

$$\Delta\phi + (\|\nabla U\|^2 - \Delta U)\phi\,. \tag{2.5}$$

This generator corresponds to the forward diffusion operator and is the adjoint of the infinitesimal generator of the backward operator, given by

$$\Delta\phi - 2\nabla\phi \cdot \nabla U\,. \tag{2.6}$$

As is well known from quantum physics, for a double well potential U, corresponding to two separated clusters, the first non-trivial eigenfunction of this operator discriminates between the two wells. This result reinforces the use of the standard graph Laplacian for computing an approximation to the normalized cut problem, as described in [13], and more generally for the use of the first few eigenvectors for spectral clustering, as suggested by Weiss [14].

In order to capture the geometry of a given manifold, regardless of the density, we propose a different normalization that asymptotically recovers the eigenfunctions of the Laplace-Beltrami (heat) operator on the manifold. For any rotation-invariant kernel $k_\varepsilon(x, y) = h(\|x-y\|^2/\varepsilon)$, we consider the normalization described in the box below. The operator

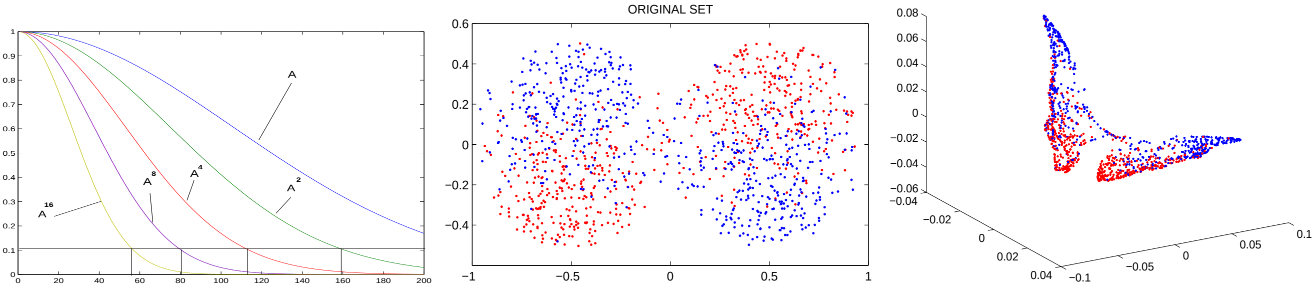


FIGURE 1. Left: spectra of some powers of A. Middle and right: consider a mixture of two materials with different heat conductivity. The original geometry (middle) is mapped as a "butterfly" set, in which the red (higher conductivity) and blue phases are organized according to the diffusion they generate: the cord length between two points in the diffusion space measures the quantity of heat that can travel between these points.

A_ε can be used to define a discrete approximate Laplace operator as follows:

$$\Delta_\varepsilon = \frac{I - A_\varepsilon}{\varepsilon},$$

and it can be verified that $\Delta_\varepsilon = \Delta_0 + \varepsilon^{\frac{1}{2}} R_\varepsilon$, where Δ_0 is a multiple of the Laplace-Beltrami operator Δ on $\mathcal{M}$, and R_ε is bounded on a fixed space of bandlimited functions. From this, we can deduce the following result:

Theorem 2.1. Let $t > 0$ be a fixed number, then as $\varepsilon \to 0$,

$$A_\varepsilon^{\frac{t}{\varepsilon}} = (I - \varepsilon\Delta_\varepsilon)^{\frac{t}{\varepsilon}} = (I - \varepsilon\Delta_0)^{\frac{t}{\varepsilon}} + \mathcal{O}(\varepsilon^{\frac{1}{2}}) = e^{-t\Delta_0} + \mathcal{O}(\varepsilon^{\frac{1}{2}}),$$

and the kernel of $A_\varepsilon^{\frac{t}{\varepsilon}}$ is given as

$$\begin{aligned} a_\varepsilon^{(\frac{t}{\varepsilon})}(x,y) &= \sum_{j\geq 0} \lambda_j^{\frac{2t}{\varepsilon}} \phi_j^{(\varepsilon)}(x)\phi_j^{(\varepsilon)}(y) \\ &= \sum_{j\geq 0} e^{-\nu_j^2 t}\phi_j(x)\phi_j(y) + \mathcal{O}(\varepsilon^{\frac{1}{2}}) \\ &= h_t(x,y) + \mathcal{O}(\varepsilon^{\frac{1}{2}}), \end{aligned}$$

where $\{\nu_j^2\}$ and $\{\phi_j\}$ are the eigenvalues and eigenfunctions of the limiting Laplace operator, $h_t(x,y)$ is the heat diffusion kernel at time t and all estimates are relative to any fixed space of bandlimited functions.

Approximation of the Laplace-Beltrami diffusion kernel

1) Let $p_\varepsilon(x) = \int_X k_\varepsilon(x,y)p(y)dy$, and form the new kernel $\hat{k}_\varepsilon(x,y) = \frac{k_\varepsilon(x,y)}{p_\varepsilon(x)p_\varepsilon(y)}$.
2) Apply the weighted graph Laplacian normalization to this kernel by defining $v_\varepsilon(x) = \int_X \hat{k}_\varepsilon(x,y)p(y)dy$, and by setting $a_\varepsilon(x,y) = \frac{\hat{k}_\varepsilon(x,y)}{v_\varepsilon(x)}$.

Then the operator $A_\varepsilon f(x) = \int_X a_\varepsilon(x,y)f(y)p(y)dy$ is an approximation of the Laplace-Beltrami diffusion kernel at time ε.

For simplicity, we assume that on the compact manifold $\mathcal{M}$, the data points are relatively densely sampled (each ball of radius $\sqrt{\varepsilon}$ contains enough sample points so that integrals can approximated by discrete sums). Moreover, if the data only covers a subdomain of $\mathcal{M}$ with nonempty boundary, then Δ_0 needs to be interpreted as acting with Neumann boundary conditions. As in the previous section, one can compute heat diffusion distances and the corresponding embedding. Moreover, any closed rectifiable curve can be embedded as a circle on which the density of points is preserved: we have thus separated the geometry of the set from the distribution of the points (see Figure 3 for an example).

2.3. **Anisotropic diffusion and stochastic differential equations.** So far we have considered the analysis of general datasets by diffusion maps, without considering the source of the data. One important case of interest is when the data x is sampled from a stochastic dynamical system. Consider therefore data sampled from a system $x(t) \in \mathbb{R}^n$ whose time evolution is described by the following Langevin equation

$$\dot{x} = -\nabla U(x) + \sqrt{2}\dot{w} \tag{2.7}$$

where U is the free energy and $w(t)$ is the standard n-dimensional Brownian motion. Let $p(y,t|x,s)$ denote the transition probability of finding the system at location y at time t, given an initial location x at time s. Then, in terms of the variables $\{y,t\}$, p satisfies the forward Fokker-Planck equation (FPE), for $t > s$,

$$\frac{\partial p}{\partial t} = \nabla \cdot (\nabla p + p\nabla U(y)) \tag{2.8}$$

while in terms of the variables $\{x,s\}$, the transition probability satisfies the backward equation

$$-\frac{\partial p}{\partial s} = \Delta p - \nabla p \cdot \nabla U(x) \tag{2.9}$$

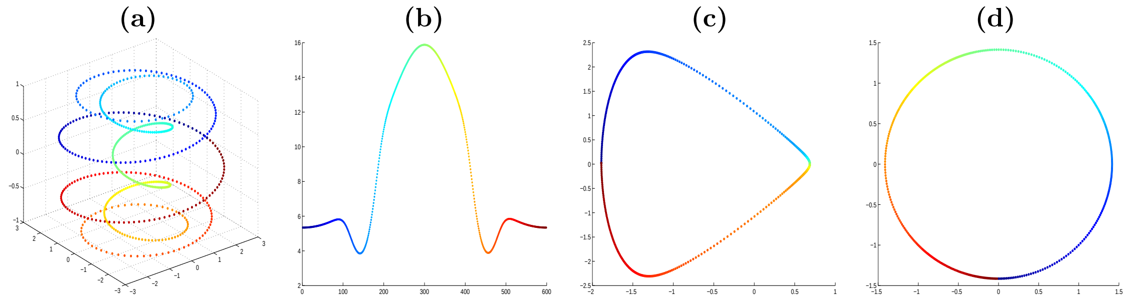


FIGURE 3. Original spiral curve (a) and the density of points on it (b), embedding obtained from the normalized graph Laplacian (c) and embedding from the Laplace-Beltrami approximation (d).

As time $t \to \infty$, the solution of the forward FPE converges to the steady state Boltzmann density

$$p(x) = \frac{e^{-U(x)}}{Z} \tag{2.10}$$

where the partition function Z is the appropriate normalization constant.

The general solution to the FPE can be written in terms of an eigenfunction expansion

$$p(x,t) = \sum_{j=0}^{\infty} a_j e^{-\lambda_j t} \phi_j(x) \tag{2.11}$$

where λ_j are the eigenvalues of the Fokker-Planck operator, with $\lambda_0 = 1 > \lambda_1 \geq \lambda_2 \geq \ldots$, and with $\phi_j(x)$ the corresponding eigenfunctions. The coefficients a_j depend on the initial conditions. A similar expansion exists for the backward equation, with the eigenfunctions of the backward operator given by $\psi_j(x) = e^{U(x)}\phi_j(x)$.

As can be seen from equation (2.11), the long time asymptotics of the solution is governed only by the first few eigenfunctions of the Fokker-Planck operator. While in low dimensions, e.g. $n \leq 3$, approximations to these eigenfunctions can be computed via numerical solutions of the partial differential equation, in general, this is infeasible in high dimensions. On the other hand, simulations of trajectories according to the Langevin equation (2.7) are easily performed. An interesting question, then, is whether it is possible to obtain approximations to these first few eigenfunctions from (large enough) data sampled from these trajectories.

In the previous section we saw that the infinitesimal generator of the normalized graph Laplacian construction corresponds to a Fokker-Planck operator with a potential $2U(x)$, see eq. (2.6). Therefore, in general, there is no direct connection between the eigenvalues and eigenfunctions of the normalized graph Laplacian and those of the underlying Fokker-Planck operator (2.8). However, it is possible to construct a different normalization that yields infinitesimal generators corresponding to the potential $U(\boldsymbol{x})$ without the additional factor of two.

Consider the following anisotropic kernel,

$$\tilde{k}_\varepsilon(x,y) = \frac{k_\varepsilon(x,y)}{\sqrt{p_\varepsilon(x)p_\varepsilon(y)}} \tag{2.12}$$

A similar analysis to that of the previous section shows that the normalized graph Laplacian construction that corresponds to this kernel gives in the asymptotic limit the correct Fokker-Planck operator, e.g., with the potential $U(x)$.

Since the Euclidean distance in the diffusion map space corresponds to diffusion distance in the feature space, the first few eigenvectors corresponding to the anisotropic kernel (2.12) capture the long-time asymptotic behavior of the stochastic system (2.7). Therefore, the diffusion map can be seen as an empirical method for homogenization (see [10] for more details). These variables are the right observables with which to implement the equation-free complex/multiscale computations of Kevrekidis *et al* (see [15] and [16]).

2.4. **One-parameter family of diffusion maps.** In the previous sections we showed three different constructions of Markov chains on a discrete data-set, that asymptotically recover either the Laplace-Beltrami operator on the manifold, or the backward Fokker-Planck operator with potential $2U(x)$ for the normalized graph Laplacian, or $U(x)$ for the anisotropic diffusion kernel.

In fact, these three normalizations can be seen as specific cases of a one-parameter family of different diffusion maps, based on the kernel

$$k_\varepsilon^{(\alpha)}(x,y) = \frac{k_\varepsilon(x,y)}{p_\varepsilon^\alpha(x)p_\varepsilon^\alpha(y)} \tag{2.13}$$

for some $\alpha > 0$.

It can be shown [9] that the forward infinitesimal operator generated by this diffusion is

$$\mathcal{H}_f^{(\alpha)}\phi = \Delta\phi - \left(e^{(1-\alpha)U}\Delta e^{-(1-\alpha)U}\right)\phi \tag{2.14}$$

One can easily see that the interesting cases are: i) $\alpha = 0$, corresponding to the classical normalized graph Laplacian, ii) $\alpha = 1$, yielding the Laplace-Beltrami operator, and iii) $\alpha = 1/2$ yielding the backward Fokker-Planck operator.

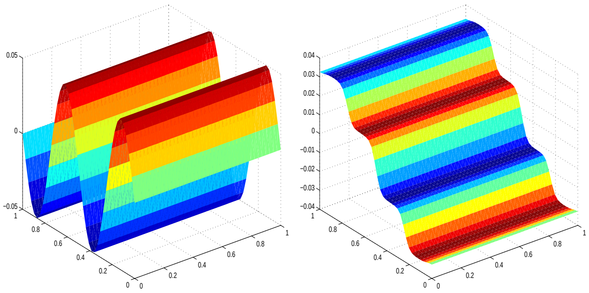


FIGURE 4. Left: the original function f on the unit square. Right: the first non-trivial eigenfunction. On this plot, the colors corresponds to the values of f.

Therefore, while the graph Laplacian based on a kernel with $\alpha = 1$ captures the geometry of the data, with the density e^{-U} playing absolutely no role, the other normalizations take into account also the density of the points on the manifold.

3. DIRECTED DIFFUSION AND LEARNING BY DIFFUSION

It follows from the previous section that the embedding that one obtains depends heavily on the choice of a diffusion kernel. In some cases, one is interested in constructing diffusion kernels which are data or task driven. As an example, consider an empirical function $F(x)$ on the data. We would like to find a coordinate system in which the first coordinate has the same level lines as the empirical function F. For that purpose, we replace the Euclidean distance in the Gaussian kernel by the anisotropic distance

$$D^2_\varepsilon(x, y) = d^2(x, y)/\varepsilon + |F(x) - F(y)|^2/\varepsilon^2$$

The corresponding limit of $A_\varepsilon^{t/\varepsilon}$ is a diffusion along the level surfaces of F from which it follows that the first nonconstant eigenfunction of A_ε has to be constant on level surfaces. This is illustrated in Figure 4, where the graph represents the function F and the colors correspond to the values of the first non-trivial eigenfunction. In particular, observe that the level lines of this eigenfunction are the integral curves of the field orthogonal to the gradient of F. This is clear since we forced the diffusion to follow this field at a much faster rate, in effect integrating that field. It also follows that any differential equation can be integrated numerically by a non-isotropic diffusion in which the direction of propagation is faster along the field specified by the equation.

We now apply this approach to the construction of empirical models for statistical learning. Assume that a data set has been generated by a process whose local statistical properties vary from location to location. Around each point x, we view all neighboring data points as having been generated by a local diffusion whose probability density is estimated by $p_x(y) = c_x \exp(-q_x(x - y))$ where q_x is a quadratic form obtained empirically by PCA from the data in a small neighborhood of x . We then use the kernel $a(x, z) = \int p_x(y)p_z(y)dy$ to model the diffusion. Note that the distance defined by this kernel is $\left(\int |p_x(y) - p_z(y)|^2 dy\right)^{1/2}$ which can be viewed as the natural distance on the "statistical tangent space" at every point in the data. If labels are available, the information about the labels can be incorporated by, for example, locally warping the metric so that the diffusion starting in one class stays in the class without leaking to other classes. This could be obtained by using local discriminant analysis (e.g. linear, quadratic or Fisher discriminant analysis) to build a local metric whose fast directions are parallel to the boundary between classes and whose slow directions are transversal to the classes (see e.g. [1]).

In data classification, geometric diffusion provides a powerful tool to identify arbitrarily shaped clusters with partially labelled data. Suppose, for example, we are given a data set X with N points from C different classes. Assume our task is to learn a function $L : X \to \{1, \dots, C\}$ for every point in X but we are given the labels of only $s << N$ points in X. If we cannot infer the geometry of the data from the label points only, many parametric methods (e.g. Gaussian classifiers) and non-parametric techniques (e.g. nearest neighbors) lead to poor results. In Figure 3, we illustrate this with an example. Here we have a hyperspectral image of pathology tissue. Each pixel (x, y) in the image is associated with a vector $\{I(x, y)\}_\lambda$ that reflects the material's spectral characteristics at different wavelengths λ. We are given a partially labelled set for three different tissue classes (marked with blue, green, and pink in 3a) and are asked to classify all pixels in the image using only spectral, as opposed to, spatial information. Both Gaussian classifiers and nearest-neighbor classifiers (see 3b) perform poorly in this case as there is a gradual change in both shading and chemical composition in the vertical direction of the tissue sample.

The diffusion framework, however, provides an alternative classification scheme that links points together by a Markov random walk (see also [17] for a discussion): let χ_i be the $\mathcal{L}^1$-normalized characteristic function of the initially labelled set from class i. At a given time t, we can interpret the diffused label functions $(A^t\chi_i)_i$ as the posterior probabilities of the points belonging to class i. Choose a time τ when the margin between the classes is maximized, and then define the label of a point $x \in X$ as the maximum *a posteriori* estimate $L(x; \tau) = \operatorname{argmax}_i A^\tau \chi_i$. Figure 3c shows the classification of the pathology sample using the above scheme. The latter result agrees significantly better with a specialist's view of correct tissue classification.

In many practical situations, the user may want to refine the classification of points that occur near the boundaries between classes in state space. One option is to use an iterative scheme, where the user provides new labelled data where needed and then restarts the diffusion with the new enlarged

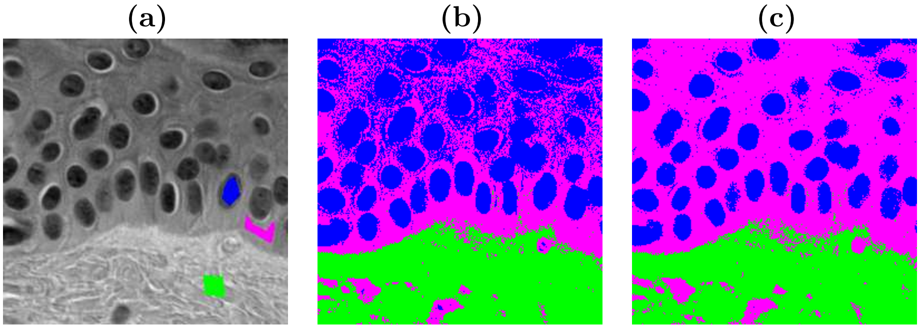


FIGURE 5. **a**: Pathology slice with partially labelled data; the 3 tissue classes are marked with blue, green and pink. **b**: Tissue classification from spectra using 1-nearest neighbors. **c**: Tissue classification from spectra using geometric diffusion.

training set. However, if the total data set X is very large, an alternative, more efficient, scheme is to define a modified kernel that incorporates both previous classification results and new information provided by the user: for example, assign to each point a score $s_i(x) \in [0,1]$ that reflects the probability that a point x belongs to class i. Then use these scores to warp the diffusion so that we have a set of class-specific diffusion kernels $\{\tilde{A}_i\}_i$ that slow down diffusion between points with different label probabilities. Choose, for example, in each new iteration, weights according to $\tilde{k}_i(x,y) = k(x,y)s_i(x)s_i(y)$ where $s_i = A^\tau \chi_i$ are the label posteriors from the previous diffusion, and renormalize the kernel to be a Markov matrix. If the user provides a series of consistent labelled examples, the classification will speed up in each new iteration and the diffusion will eventually occur only within disjoint sets of samples with the same labels.

4. SUMMARY

In this paper, we presented a general framework for structural multiscale geometric organization of graphs and subsets of $\mathbb{R}^n$. We introduced a family of diffusion maps that allow the exploration of both the geometry, the statistics and functions of the data. Diffusion maps provide a natural low-dimensional embedding of high-dimensional data that is suited for subsequent tasks such as visualization, clustering, and regression. In part II of this paper, we introduce multiscale methods that allow fast computation of functions of diffusion operators on the data. We also present a scheme for extending empirical functions.

5. ACKNOWLEDGMENTS

The authors would like to thank Ioannis Kevrekidis for very helpful suggestions and discussions and Naoki Saito for his useful comments during the preparation of the manuscript. This work was partially supported by DARPA and AFOSR.

REFERENCES

[1] Hastie, T., Tibshirani, R. & Friedman, J.H. (2001) *The Elements of Statistical Learning*, Springer-Verlag, pp. 144-155.

[2] Coifman, R.R. & Saito, N. (1994) Constructions of Local Orthonormal Bases for Classification and Regression, *Comptes Rendus de l'Académie des Sciences, Paris, Série I* **319**, pp. 191-196.

[3] Ham, J., Lee, D.D., Mika, S., & Schölkopf, B. (2003) A Kernel View of the Dimensionality Reduction of Manifolds, Tech. report TR-110 (Max-Planck-Institut für Biologische Kybernetik), pp. 1-9.

[4] Roweis, S.T. & Saul, L.K. (2000) Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science* **290**, pp. 2323-2326.

[5] Belkin, M. & Niyogi, P. (2003) Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, *Neural Computation* **15**, pp. 1373-1396.

[6] Donoho, D.L. & Grimes, C. (2003) Hessian Eigenmaps: New Locally Linear Embedding Techniques for High-Dimensional Data, *Proceedings of the National Academy of Sciences* **100**, pp. 5591-5596.

[7] Zhang, Z. & Zha, H. (2002) Principal Manifolds and Nonlinear Dimension Reduction via Local Tangent Space Alignement, Tech. report CSE-02-019 (Department of Computer Science and Engineering, Pennsylvania State University), pp. 1-22.

[8] Chung, F.R.K. (1997) *Spectral Graph Theory*, Conference Board of the Mathematical Sciences, American Mathematical Society.

[9] Coifman, R.R. & Lafon, S. (2004) Diffusion Maps, *Applied and Computational Harmonic Analysis*, in press.

[10] Nadler, B., Lafon, S., Coifman, R.R. & Kevrekidis, I., (2004) Diffusion Maps, Spectral Clustering and Reaction Coordinates of Dynamical Systems, *Applied and Computational Harmonic Analysis*, in press.

[11] Pedersen, K.S. & Lee, A.B. (2002) Toward a Full Probability Model of Edges in Natural Images, *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, Part I* (Nielsen, M., Heyden, A., Sparr, G. & Johansen, P. eds.), Springer, pp.328-342.

[12] Huggins, P.S., and Zucker, S.W. (2002) Representing Edge Models via Local Principal Component Analysis, *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, Part I* (Nielsen, M., Heyden, A., Sparr, G. & Johansen, P. eds.), Springer, pp.384-398.

[13] Shi, J. & Malik, J. (2000) Normalized Cuts and Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, pp. 888-905.

[14] Weiss, Y. (1999) Segmentation Using Eigenvectors: a Unifying View, *Proceedings of the Institute of Electrical and Electronics Engineers International Conference on Computer Vision*, pp. 975-982.

[15] Gear, C.W., Kevrekidis, I.G. & Theodoropoulos, C. (2002) "Coarse" Integration/Bifurcation Analysis via Microscopic Simulators: Micro-Galerkin Methods, *Computers and Chemical Engineering* **26**, pp. 941-963.

[16] Kevrekidis, I.G., Gear, C.W., Hyman, J.M., Kevrekidis, P.G., Runborg, O. & Theodoropoulos, K. (2003) Equation-Free Coarse-Grained Multiscale Computation: Enabling Microscopic Simulators to Perform System-Level Tasks, *Communications in Mathematical Sciences* **1**, pp. 715-762.

[17] Szummer, M. & Jaakkola, T. (2001) Partially Labeled Classification with Markov Random Walks, *Advances in Neural Information Processing Systems* **14** (Dietterich, T.G., Becker. S. & Ghahramani, Z. eds.), MIT press.

Department of Mathematics, Program in Applied Mathematics, Yale University, 10 Hillhouse Ave, New Haven, CT, 06510

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

GEOMETRIC DIFFUSIONS AS A TOOL FOR HARMONIC ANALYSIS AND STRUCTURE DEFINITION OF DATA

PART II: MULTISCALE METHODS

R.R.Coifman [1], S.Lafon [1], A.B.Lee [1], M.Maggioni[1], B.Nadler [1], F.J.Warner [1], S.W.Zucker [2]

Abstract. In the companion paper a framework for structural multiscale geometric organization of subsets of $\mathbb{R}^n$ and of graphs was introduced. Here diffusion semigroups are used to generate multiscale analyses in order to organize and represent complex structures. We emphasize the multiscale nature of these problems, and we build scaling functions of Markov matrices (describing local transitions) that lead to macroscopic descriptions at different scales. The process of iterating or diffusing the Markov matrix is seen as a generalization of some aspects of the Newtonian paradigm, in which local infinitesimal transitions of a system lead to global macroscopic descriptions by integration. This part deals with the construction of fast order N algorithms for data representation and for homogenization of heterogeneous structures.

1. Introduction

In the companion paper [1] it is shown that the eigenfunctions of a diffusion operator A can be used to perform global analysis of the set and of functions on a set. Here we present a construction of a multiresolution analysis of functions on the set related to the diffusion operator A. This allows to perform a local analysis at different diffusion scales.

This is motivated by the fact that in many situations one is interested not in the data itself, but in functions on the data, and in general these functions exhibit different behaviour at different scales. This is the case in many problems in learning, in analysis on graphs, in dynamical systems etc... The analysis through the eigenfunctions of Laplacian considered in [1] are global and are affected by global characteristics of the space. It can be thought of as global Fourier analysis. The multiscale analysis proposed here is in the spirit of wavelet analysis.

We refer the reader to [2, 3, 4] for further details and applications of this construction, as well as a discussion of the many relationships between this work and the work of many other researchers in several branches of mathematics and applied mathematics. Here we would like to at least mention the relationship with Fast Multipole Methods [5, 6], Algebraic Multigrid [7], lifting [8, 9].

2. Multiscale Analysis of Diffusion

2.1. **Construction of the Multiresolution Analysis.** Suppose we are given a self-adjoint diffusion operator A as in [1] acting on $\mathcal{L}^2$ of a metric measure space (X, d, μ). We interpret A as a *dilation* operator, and use it to define a multiresolution analysis. It is natural to discretize the semigroup $\{A^t\}_{t\geq 0}$ of the powers of A at a logarithmic scale, for example at the times

$$t_j = 1 + 2 + 2^2 + ... + 2^j = 2^{j+1} - 1 \tag{2.1}$$

For a fixed $\epsilon \in (0, 1)$, we define the approximation spaces by

$$V_j = \overline{< \{\phi_i : \lambda_i^{t_j} \geq \epsilon\} >} \tag{2.2}$$

where the ϕ_i's are the eigenvectors of A, ordered by decreasing eigenvalue. We will denote by P_j the orthogonal projection onto V_j. The set of subspaces $\{V_j\}_{j\in\mathbb{Z}}$ is a multiresolution analysis in the sense that it satisfies the following properties:

(i) $\lim_{j\to-\infty} V_j = \mathcal{L}^2(X, \mu)$,
$\lim_{j\to+\infty} V_j = \overline{< \{\phi_i : \lambda_i = 1\} >}$.
(ii) $V_{j+1} \subseteq V_j$ for every $j \in \mathbb{Z}$.
(iii) $\{\phi_i : \lambda_i^{t_j} \geq \epsilon\}$ is an orthonormal basis for V_j.

We can also define the detail subspaces W_j as the orthogonal complement of V_j in V_{j+1}, so that we have the familiar relation between approximation and detail subspaces as in the classical wavelet multiresolution constructions:

$$V_{j+1} = V_j \oplus^{\perp} W_j.$$

This is very much in the spirit of a Littlewood-Paley decomposition induced by the diffusion semigroup [10]. However, in each subspace V_j and W_j we have the orthonormal basis of eigenfunctions, but we would like to replace them with localized orthonormal bases of scaling functions as in wavelet theory. Generalized Heisenberg principles (see also section 4) put a lower bound on how much localization can be achieved at each scale j, depending on the spectrum of the operator A and on the space on which it acts. We would like to have basis elements as much localized as allowed by the Heisenberg principle at each scale, and spanning (approximately) V_j. We do all this while avoiding computation of the eigenfunctions.

We start by fixing a precision $\epsilon > 0$, and assume that A is represented on the basis $\Phi_0 = \{\delta_k\}_{k\in X}$. We consider the columns of A, which can be interpreted as the set of functions $\tilde{\Phi}_1 = \{A\delta_k\}_{k\in X}$ on X. We use a local multiscale Gram-Schmidt procedure, described below, to carefully but efficiently orthonormalize these columns into a basis $\Phi_1 = \{\varphi_{1,k}\}_{k\in X_1}$ (X_1 is *defined* as this index set) for the range of

[1] Department of Mathematics, Yale University, 10 Hillhouse Ave, New Haven, CT, 06510, U.S.A., +1-(203)-432-1278

[2] Department of Computer Science, Yale University, 51 Prospect St., New Haven, CT, 06510, U.S.A, +1-(203)-432-1278

A up to precision ϵ. This is a linear transformation we represent by a matrix G_0. This yields a subspace that is ϵ-close to V_1. Essentially Φ_1 is a basis for a subspace which is ϵ-close to the range of A, the basis elements that are well-localized and orthogonal. Obviously $|X_1| \leq |X|$ but the inequality may already be strict since part of the range of A may be below the precision ϵ. Whether this is the case or not, we have then a map M_0 from X to X_1, which is the composition of A with the orthonormalization by G_0. We can also represent A in the basis Φ_1: we denote this matrix by A_1 and compute A_1^2. See the diagram in Figure 1.

We now proceed by looking at the columns of A_1^2, which are $\tilde{\Phi}_2 = \{A_1^2\delta_k\}_{k\in X_1} = \{A^2\varphi_{1,k}\}_{k\in X_1}$ up to precision ϵ, by unravelling the bases on which the various elements are represented. Again we can apply a local Gram-Schmidt procedure to orthonormalize this set: this yields a matrix G_1 and an orthonormal basis $\Phi_2 = \{\varphi_{2,k}\}_{k\in X_2}$ for the range of A_1^2 up to precision ϵ, and hence for the range of A_0^3 up to precision 2ϵ. Moreover, depending on the decay of the spectrum of A, $|X_2| << |X_1|$. The matrix M_1 which is the composition of G_1 with A_1^2 is then of size $|X_2| \times |X_1|$, and $A_2^2 = M_1 M_1^T$ is a representation of A^4 acting on Φ_2.

After j steps in this fashion, we will have a representation of $A^{1+2+2^2+\dots+2^j} = A^{2^{j+1}-1}$ onto a basis $\Phi_j = \{\varphi_{j,k}\}_{k\in X_j}$, that spans a subspace which is $j\epsilon$-close to V_j. Depending on the decay of the spectrum of A, we expect $|X_j| << |X|$, in fact in the ideal situation[3] the spectrum of A decays fast enough so that there exists $\gamma < 1$ such that $|X_j| < \gamma^{2^{j+1}-1}|X|$. This subspace is spanned by "bump" functions at scale j, as defined by the corresponding power of the diffusion operator A. The "centers" of these bump functions can be identified with X_j, which we can think of X_j as a coarser version of X. The basis Φ_j is naturally identified with the set of Dirac δ-functions on X_j, however can extend these functions, defined on the "compressed" graph X_j to the whole initial graph X by writing

$$\begin{aligned}\varphi_{j,k}(x) &= M_{j-1}\varphi_{j-1,k}(x) \quad ,x \in X_{j-1} \\ &= M_{j-1}M_{j-2}\cdot\ \dots\ \cdot M_0\,\varphi_{0,k}(x)\ ,x \in X_0\,.\end{aligned} \tag{2.3}$$

Since every function in Φ_0 is defined on X, so is every function in Φ_j. Hence any function on the compressed space X_j can be extended naturally to the whole X. In particular, one can compute low-frequency eigenfunctions on X_j, and then extend them to the whole X. This is of course completely analogous to the standard construction of scaling functions in the Euclidean setting [11, 5, 12].Observe that each point in X_j can be considered as a "local aggregation" of points in X_{j-1}, which is completely dictated by the action of the operator A on functions on X: A itself is dictating the geometry with respect to which it should be analyzed, compressed, applied to any vector.

$$\begin{tikzcd}
\Phi_0 \arrow[r, "M_0"] \arrow[d, "A_0"'] & \Phi_1 \arrow[r, "M_1"] \arrow[d, "A_1^2"'] & \dots \arrow[r, "M_j"] & \Phi_{j+1} \arrow[r, "M_{j+1}"] \arrow[d, "A_j^{2^j}"'] & \dots \\
\tilde{\Phi}_1 \arrow[ur, "G_0"] & \tilde{\Phi}_2 \arrow[ur, "G_1"] & \dots \arrow[ur, "G_j"] & \tilde{\Phi}_{j+1} \arrow[ur, "G_{j+1}"] &
\end{tikzcd}$$

Figure 1. Diagram for downsampling, orthogonalization and operator compression.

We have thus computed and efficiently represented the powers A^{2^j}, for $j > 0$, which describe the behaviour of the diffusion at different time scales. This applies to the solution of discretized of partial differential equations, of Markov chains, and in learning and related classification problems.

2.2. Wavelet transforms and Green's function.

The construction immediately suggests an associated fast scaling function transform: suppose we are given f on X and want to compute $< f, \varphi_{j,k} >$ for all scales j and corresponding "translations" k. Being given f is equivalent to saying we are given $(< f, \varphi_{0,k} >)_{k\in X}$. Then we can compute $(< f, \varphi_{1,k} >)_{k\in X_1} = M_0(< f, \varphi_{0,k} >)_{k\in X}$, and so on for all scales. The matrices M_j are sparse (since A_j and G_j are), so this computation is fast. This generalizes the classical scaling function transform. We will see later that wavelets can be constructed as well and a fast wavelet transform is possible.

In the same way, any power of A can be applied fast to a function f. In particular the Green's function $(I - A)^{-1}$ can be applied fast to any function: since

$$(I - A)^{-1}f = \sum_{k=1}^{+\infty} A^k f\,,$$

if we let $S_K = \sum_{k=1}^{2^K} A^k$ we see that

$$S_{K+1} = S_K + A^{2^K} S_K = \prod_{k=0}^{K}\left(I + A^{2^k}\right)f\,,$$

and each term of the product can be applied fast to f.

The construction of the multiscale bases can be done in time $\mathcal{O}(n\log^2 n)$, where $n = |X|$, if the spectrum of A has fast enough decay. The decomposition of a function f onto the scaling functions and wavelets we construct can be done in the same time, and so does the computation of $(I - A)^{-1}f$.

2.3. The orthogonalization process.

We sketch here how the orthogonalization works: for details refer to [3, 2]. Suppose we start from a δ-local basis $\Phi = \{\varphi_x\}_{x\in\mathcal{T}}$ (in our case, φ_x is going to be a bump $A^l\delta_x$). We greedily build a first layer of basis functions $\Phi_0 = \{\tilde{\varphi}_{0,x_k}\}_{x_k\in\mathcal{K}_0}$, $\mathcal{K}_0 \subseteq \mathcal{T}$ as follows. We let φ_{0,x_0} be a basis function with greatest $\mathcal{L}^2$-norm. Then we let φ_{0,x_1} be a basis function with biggest $\mathcal{L}^2$-norm among the basis functions with support disjoint from the support of φ_{0,x_0} but not farther than δ from it. By induction, after $\varphi_{0,x_0}, \dots, \varphi_{0,x_l}$ have been chosen, we let $\varphi_{0,x_{l+1}}$ be a

[3] By Weyl's Theorem on the distribution function of the spectrum of the Laplace-Beltrami operator, this is the case when A is an accurate enough discretization of the Laplace-Beltrami on a smooth compact Riemannian manifold with smooth boundary.

scaling function with largest $\mathcal{L}^2$-norm among those having a support which does not intersect any of the supports of the basis functions already constructed, but is not farther than δ from the closest such support. We stop when no such choice can be made. One can think of $\mathcal{K}_0$ roughly as a 2δ lattice.

At this point Φ_0 in general spans a subspace much smaller than the one spanned by Φ. We construct a second layer $\Phi_1 = \{\tilde{\varphi}_{1,x_k}\}_{x_k\in\mathcal{K}_1}$, $\mathcal{K}_1 \subseteq \mathcal{T} \setminus \mathcal{K}_0$ as follows. Orthogonalize each $\{\varphi_x\}_{x\in\mathcal{T}\setminus\mathcal{K}_0}$ to the functions $\{\varphi_{0,x_k}\}_{x_k\in\mathcal{K}_0}$. Observe that since the support of φ_x is small, this orthogonalization is local, in the sense that each φ_x needs to be orthogonalized only to the few $\varphi'_{0,x_k}s$ that have an intersecting support. In this way we get a set $\tilde{\Phi}_1$, orthogonal to Φ_0 but not orthogonal itself. We orthonormalize it exactly as we did to get Φ_0 from Φ. We proceed by building as many layers as necessary to span the whole space $< \Phi >$ (up to the specified precision ϵ).

2.4. **Wavelets.** We would like to construct bases $\{\psi_{j,k}\}_k$ for the spaces W_j, $j \geq 1$, such that $V_j \oplus^{\perp} W_j = V_{j+1}$. To achieve this, after having built $\{\varphi_{j,k}\}_{k\in\mathcal{K}_j}$ and $\{\varphi_{j+1,k}\}_{k\in\mathcal{K}_{j+1}}$, we can apply our modified Gram-Schmidt procedure with geometric pivoting to the set of functions

$$\{(P_j - P_{j+1})\varphi_{j,k}\}_{k\in\mathcal{K}_j},$$

which will yield an orthonormal basis of wavelets for the orthogonal complement of V_{j+1} in V_j. Observe that each wavelet is a result of an orthogonalization process which is local, so the computation is again fast. To achieve numerical stability we orthogonalize at each step the remaining $\varphi_{j+1,k}$'s to both the wavelets built so far and $\varphi_{j,k}$. Wavelet subspaces can be recursively split further to obtain diffusion wavelet packets [4], which allow the application of the classical fast algorithms [13] for denoising [14], compression [15] and discrimination [16].

3. Examples and applications

Example 3.1 (Multiresolution diffusion on the homogeneous circle). To compare with classical constructions of wavelets, we consider the unit circle, sampled at 256 points, and the classical isotropic heat diffusion on it. The initial orthonormal basis Φ_0 is given by the set of δ-functions at each point, and we build the diffusion wavelets at all scales, which clearly relate to splines and multiwavelets. The spectrum of the diffusion operator does not decay very fast. See Figure 2 and 3.

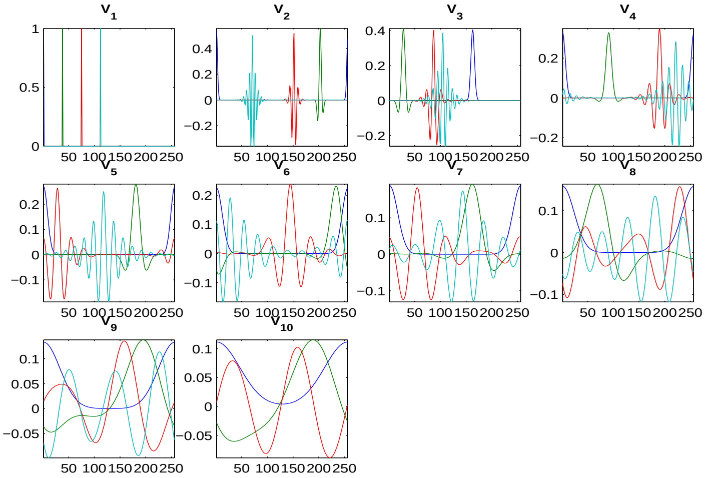


Figure 2. Diffusion Multiresolution Analysis on the circle. We consider 256 points on the unit circle, starting with $\varphi_{0,k} = \delta_k$ and with the standard diffusion. We plot several scaling functions in each approximation space V_j.

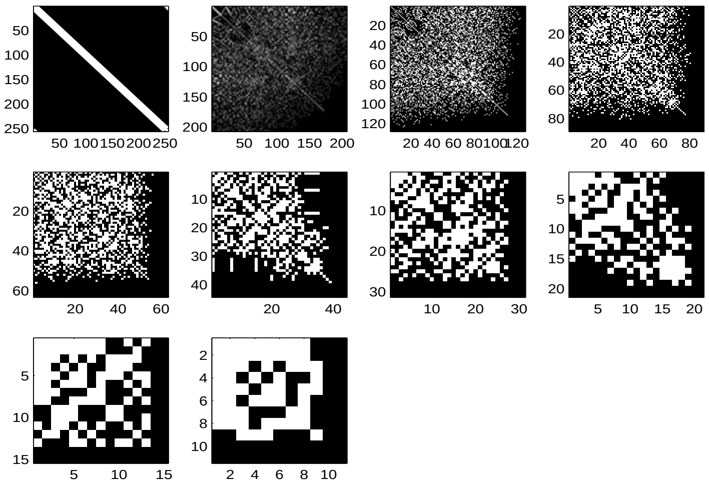


Figure 3. Diffusion Multiresolution Analysis on the circle: we plot the compressed matrices representing powers of the diffusion operator, in white are the entries above working precision (here set to 10^{-8}). Notice the shrinking of the size of the matrices which are being compressed at the different scales.

Example 3.2 (Dumbbell). We consider a dumbbell-shaped manifold, sampled at 1400 points, and the diffusion associated to the (discretized) Laplace-Beltrami operator as discussed in [1]. See Figure 4 for the plots of some scaling functions and wavelets: they exhibit the expected locality and multiscale features, dependent on the intrinsic geometry of the manifold.

Example 3.3 (Multiresolution diffusion on a nonhomogenous circle). We can apply the construction of diffusion wavelets to non-isotropic diffusions arising from partial differential equations, to tackle problems of homogenization in a natural way. The literature on homogenization is vast, see e.g. [17, 18, 19, 20, 21] and references therein.

Our definition of scales which is driven by the differential operator, which in general results in highly nonuniform and nonhomogeneous spatial and spectral scales, and in corresponding coarse equations of the system, which have high precision.

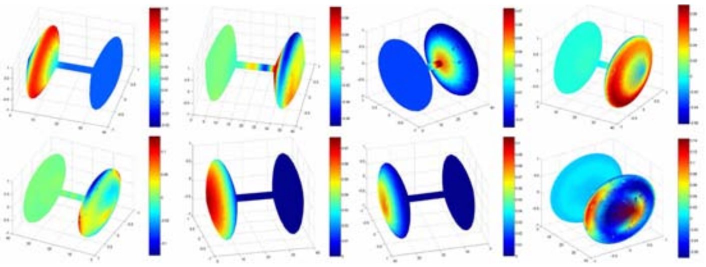


FIGURE 4. Some diffusion scaling functions and wavelets at different scales on a dumbbell-shaped manifold sampled at 1400 points.

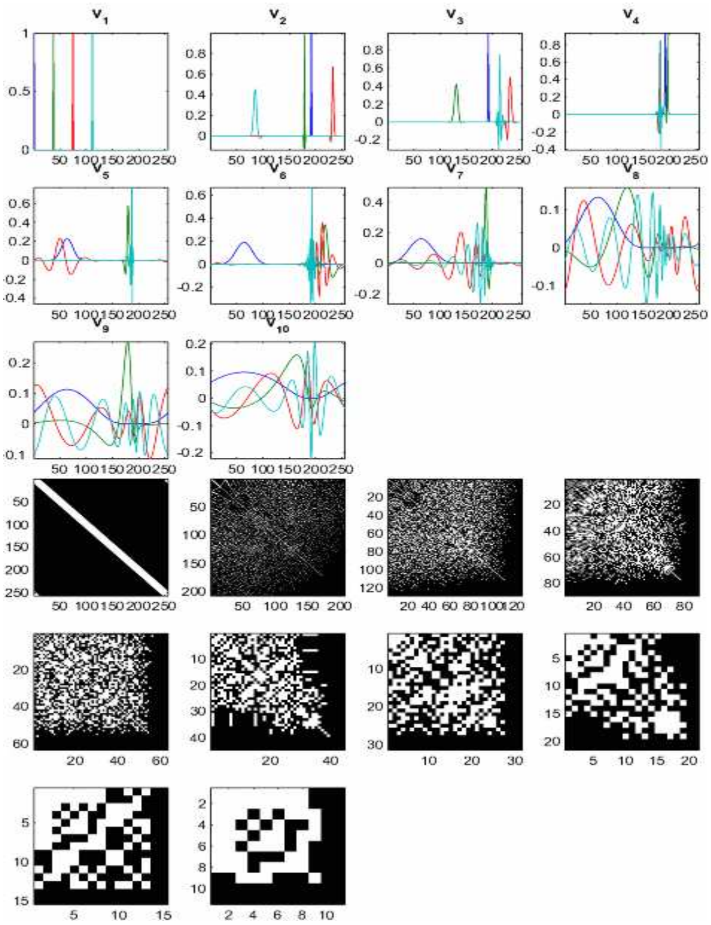


FIGURE 5. Multiresolution Diffusion on a circular medium with non-constant diffusion coefficient. Top: several scaling functions and wavelets in different approximation subspaces V_j: notice that scaling functions at the same diffusion scale exhibit different spatial localization, which depends on the local diffusion coefficient. Bottom: matrix compression of the dyadic powers of T on the scaling function bases of the V_j's: notice the size of the matrices shrinking with scale.

For example we can consider the non-homogeneous heat equation on the circle

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(c(x)\frac{\partial}{\partial x}u\right) \tag{3.1}$$

where $c(x)$ is a positive function close 0 at certain points and almost 1 at others. We want to represent the intermediate and large scale/time behavior of the solution by compressing powers of the operator representing the discretization of the spatial differential operator $\frac{\partial}{\partial x}\left(c(x)\frac{\partial}{\partial x}\right)$. The *spatial* differential operator on the right-hand side of (3.1) is a matrix T which, when properly normalized, can be interpreted as a non-translation invariant random walk. Our construction yields a multiresolution associated to this operator that is highly nonuniform, with most scaling functions concentrated around the points where the conductivity is highest, for several scales. The compressed matrices representing the (dyadic) powers of this operator can be viewed as multiscale homogenized versions, at a certain scale which is *time and space* dependent, of the original operator, see Figure 5.

While the examples above illustrate classical settings, the construction of diffusion wavelets carries over unchanged to weighted graphs, by considering the generator of the diffusion associated to the natural random walk (and Laplacian) on the graph. It then allows a natural multiscale analysis of functions of interest on such a graph. We expect this to have a wide range of applications to the analysis of large data sets, document corpora, network traffic, et al., which are naturally modelled by graphs.

4. EXTENSION OF EMPIRICAL FUNCTIONS OFF THE DATA SET

An important aspect of the multiscale developed so far involves the relation of the spectral theory on the set to the localization on and off the set of the corresponding eigenfunctions and diffusion scaling functions and wavelets. In addition to the theoretical interest of this topic, the extension of functions defined on a set X to a larger set $\overline{X}$ is of critical importance in applications such as statistical learning. To this end, we construct a set of functions, termed *geometric harmonics*, that allow to extend a function f off the set X, and we explain how this provides a multiscale analysis of f. For a more detailed studied of geometric harmonics, the reader is referred to [22].

4.1. **Construction of the extension: the geometric harmonics.** Let's specify the mathematical setting. Let X be a set contained in a larger set $\overline{X}$, and μ be a measure on X. Suppose that one is given a positive semi-definite symmetric kernel $k(\cdot,\cdot)$ defined on $\overline{X}\times\overline{X}$, and if f is defined on X, let $K : L^2(X,\mu)\to L^2(X,\mu)$ be defined by

$$Kf(x) = \int_X k(x,y)f(y)d\mu(y)\,.$$

Let $\{\psi_j\}$ and $\{\lambda_j^2\}$ be the eigenfunctions and eigenvalues of this operator. Note that under weak hypotheses, the operator K is compact, and its eigenfunctions form a basis of $L^2(X,\mu)$. Then by definition, if $\lambda_j^2 > 0$, then

$$\psi_j(x) = \frac{1}{\lambda_j^2}K\psi_j(x) = \frac{1}{\lambda_j^2}\int_X k(x,y)\psi_j(y)d\mu(y)\,,$$

where this identity holds for $x \in X$. Now if we let x be in $\overline{X}$, the right-hand side of this equation is well-defined, and this allows to extend ψ_j as a function $\overline{\psi}_j$ defined on $\overline{X}$. This procedure, that goes by the name of Nyström extension, has been already suggested to overcome the problem of large scale data sets [23], and to speed up the data processing [24].

From the above, each extension is constructed as an integral of the values over the smaller set X, and consequently verifies some sort of mean value theorem. We call these functions *geometric harmonics*.

From the numerical analysis point of view, one has to be careful as $\lambda_j \to 0$ as $j \to +\infty$, and one can extend only the eigenfunctions ψ_j for which $\lambda_j^2 > \delta\lambda_0^2$, where $\delta > 0$ is preset number. We can now safely define the extension of function f from X to $\overline{X}$ by

$$\overline{f}(x) = \sum_{\lambda_j^2 > \delta\lambda_0^2} \langle \psi_j, f\rangle_X \overline{\psi}_j(x)$$

for $x \in \overline{X}$, where $\langle\cdot,\cdot\rangle_X$ is the inner product of $L^2(X,\mu)$. This way, the extension operation has condition number $\frac{1}{\delta}$. We immediately notice that for $\overline{f}$ to approximately coincide with f on X, one must have that most of the energy of f be concentrated in the first few eigenfunctions ψ_j.

Let's give three examples of geometric harmonics. The first example is related to potential theory. Assume that X is a smooth closed hypersurface of $\mathbb{R}^n = \overline{X}$, $d\mu = dx$ and consider the Newtonian potential in $\mathbb{R}^n$:

$$k(x,y) = \begin{cases} -\log(\|x-y\|) & \text{if } n = 2,, \\ \frac{1}{\|x-y\|^{n-2}} & \text{if } n \geq 3\,. \end{cases}$$

Then the geometric harmonics have the form

$$\overline{\psi}_j(x) = \frac{1}{\lambda_j^2}\int_X k(x,y)\psi_j(y)dy\,,$$

and are obviously harmonic in the domain with boundary X. If f is a function on X representing the single layer density of charges on X, then the extension $\overline{f}$ is, by construction, a sum of harmonic functions, and is an harmonic extension of f.

For the second example, consider a Hilbert basis $\{e_j\}_{j\in\mathbb{Z}}$ of a subspace V of $L^2(\mathbb{R}^n) \cap C(\mathbb{R}^n)$. For instance, this could be a wavelet basis of some finite scale shift-invariant space. Then the diagonalization of the restriction of kernel

$$k(x,y) = \sum_{j\in\mathbb{Z}} e_n(x)e_n^*(y)$$

to a set X generates geometric harmonics, and an extension procedure of empirical functions on X to functions of V.

The third example is of particular importance as it generalizes the Prolate Spheroidal Wave Functions introduced in the context of signal processing by [25, 26]. Assume that $X \subset \mathbb{R}^n$ and consider the space V_B of bandlimited functions with fixed band $B > 0$ (we call these functions $B-$bandlimited). Following the procedure explained in the second example, we can construct geometric harmonics $\{\overline{\psi}_j\}$ that are $B-$bandlimited. It can be shown that this comes down to diagonalizing the kernel

$$k_B(x,y) = \int_{\|\xi\|<B} e^{2i\pi\langle\xi,x\rangle}e^{-2i\pi\langle\xi,y\rangle}d\xi = \frac{J_{\frac{n}{2}}(2\pi B\|x-y\|)}{\|x-y\|^{\frac{n}{2}}}$$

where x and y belong to X, and J_ν is the Bessel function of the first type and of order ν. From the first equality sign, we see that the geometric harmonics arise from a Principal Component Analysis of the set of all restrictions of $B-$bandlimited complex exponentials to X.

It can verified that, in addition to be orthogonal on the set X, these $B-$bandlimited geometric harmonics are also orthogonal over the whole space $\mathbb{R}^n$. Moreover, $\overline{\psi}_j$ minimizes the Rayleight quotient

$$\frac{\int_{\mathbb{R}^n} |\overline{f}(x)|^2dx}{\int_X |f(x)|^2dx}$$

under the constraint that f be orthogonal to $\{\psi_0, \psi_1, ..., \psi_{j-1}\}$. In other words, $\overline{\psi}_0$ is the $B-$bandlimited extension of ψ_j that has minimal energy on $\mathbb{R}^n$. As a consequence, $\overline{f}$ is the $B-$bandlimited extension of f that has minimal energy off the set X. This type of extension is optimal in the sense that it is the average of all $B-$bandlimited extension of f. It also suggests that this extension satisfies Occam's razor in that it is the "simplest" among all bandlimited extensions: any other extension is equal to $\overline{f}$ plus an orthogonal bandlimited function that vanishes on X.

4.2. **Multiscale extension.** For a given function f on X, we have constructed a minimal energy $B-$bandlimited extension $\overline{f}$. In the case when X is a smooth compact submanifold of $\mathbb{R}^n$, we can now relate the spectral theory on the set X to that on $\mathbb{R}^n$.

On the one hand, any band limited function of band $B > 0$ restricted to X can be expanded to exponential accuracy in terms of the eigenfunctions of the Laplace-Beltrami operator Δ with eigenvalues ν_j^2 not exceeding CB^2 for some small constant $C > 0$. On the other hand, it can be shown that every eigenfunction of the Laplace-Beltrami operator satisfying this condition extends as a bandlimited function with band $C'B$. Both of these statements can be proved by observing that eigenfunctions on the manifold are well approximated by restrictions of bandlimited functions.

We conclude that any empirical function f on X that can be approximated as a linear combination of eigenfunctions of Δ, and these eigenfunctions can be extended to different distances: if the eigenvalue is ν^2, then the corresponding eigenfunction can be extended as a $\nu-$bandlimited function off the set X to a distance $C\nu^{-1}$. This observation constitutes a formulation of the Heisenberg principle involving the Fourier analysis on and off the set X, and which states that any empirical function can be extended as a sum of "atoms" whose numerical supports in the ambient space is related to their frequency content on the set.

The generalized Heisenberg principle is illustrated on figure 4.2, where we show the extension of the functions $f_j(\theta) =$

$\cos(2\pi j\theta)$ for $j = 1, 2, 3$ and 4, from the unit circle to the plane. For each function, we used gaussian kernels, and the scale was adjusted as the maximum scale that would preserve a given accuracy.

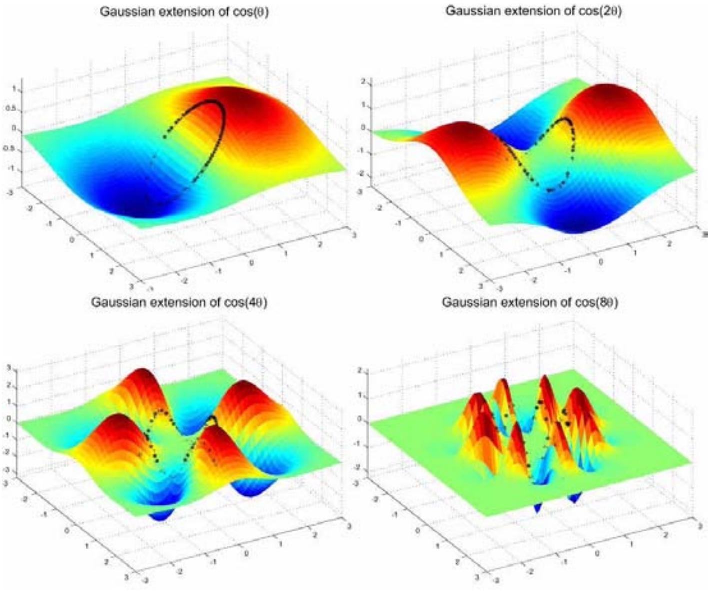


FIGURE 6. Extension of the functions $f_j(\theta) = \cos(2\pi j\theta)$ for $j = 1, 2, 3$ and 4, from the unit circle to the plane.

5. Conclusion

We have introduced a multiscale structure for the efficient computation of large powers of a diffusion operator, and its Green's function, based on a generalization of wavelets to the general setting of discretized manifolds and graphs. This has application to the numerical solution of partial differential equations, and to the analysis of functions on large data sets and learning. We have shown that a global (with eigenfunctions of the Laplacian) or local (with diffusion wavelets) analysis on a manifold embedded in Euclidean space can be extended outside the manifold in a multiscale fashion using band-limited functions.

6. Acknowledgments

The authors would like to thank James C. Bremer Jr., Naoki Saito, Raanan Schul, and Arthur D. Szlam for their useful comments and suggestions during the preparation of the manuscript. This work was partially funded by DARPA and AFOSR.

References

[1] Coifman, R.R., Lafon, S., Lee,A.B., Maggioni, M., Nadler, B., Warner, F.J. and Zucker,S.W. (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data. part i: diffusion maps. *Proc. Natl. Acad. Sci.* USA, in press.

[2] Coifman, R.R., and Maggioni, M. (2004) Multiresolution analysis associated to diffusion semigroups: construction and fast algorithms. *Tech. Rep. YALE/DCS/TR-1289, Dept. Comp. Sci., Yale Univ.*.

[3] Coifman, R.R., and Maggioni, M. (2004) Diffusion wavelets. *Tech. Rep. YALE/DCS/TR-1303, Yale Univ.* and *Appl. Comp. Harm. Anal.*, in press.

[4] Bremer, J.C. Jr., Coifman, R.R., Maggioni, M., and Szlam, A.D. (2004) Diffusion wavelet packets. *Tech. Rep. YALE/DCS/TR-1304, Yale Univ.* and *Appl. Comp. Harm. Anal.*, in press.

[5] Beylkin, G., Coifman, R.R., and Rokhlin, V. (1991) Fast wavelet tranforms and numerical algorithms. *Comm Pure Applied math*, **44**,141–183.

[6] Greengard, L. and Rokhlin, V. (1987) A fast algorithm for particle simulations. *J Comput Phys*, **73**,325–348.

[7] Brandt, A. (1986) Algebraic multigrid theory: the symmetric case. *Appl. Math. Comp.*, **19**,23–56.

[8] Sweldens, W. (1996) The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, **3**(2),186–200.

[9] Sweldens, W. (1997) The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, **29**(2),511–546.

[10] Stein, E. (1970) *Topics in Harmonic Analysis related to the Littlewood-Paley theory.* **63**. Princeton University Press.

[11] Meyer, Y. (1990) *Ondelettes et Operatéurs.* Hermann, Paris.

[12] Daubechies, I. (1992) *Ten lectures on wavelets.* Society for Industrial and Applied Mathematics.

[13] Coifman, R.R., Meyer, Y., Quake, S., and Wickerhauser, MV (1993) Signal processing and compression with wavelet packets. In *Progress in wavelet analysis and applications (Toulouse, 1992)*, 77–93. Frontières, Gif.

[14] Donoho, D.L. and Johnstone, I.M. (1994) Ideal denoising in an orthonormal basis chosen from a library of bases. Technical report, Stanford University.

[15] Coifman, R.R. and Wickerhauser, M.V. (1992) Entropy-based algorithms for best basis selection. *IEEE Trans. Info. Theory*, **38**,713–718.

[16] Coifman, R.R. and Saito, N. (1994) Constructions of local orthonormal bases for classification and regression. *C. R. Acad. Sci. Paris*, **319** Série I,191–196.

[17] Babuska, I. (1976) *Numerical Solutions of Partial Differential Equations - III*, Homogenization and its application. 89–116. Ed. New York: Academic.

[18] Gilbert, A.C. (1998) A comparison of multiresolution and classical one-dimensional homogenization schemes. *Appl. Comp. Har. Anal.*, **5**,1–35.

[19] Beylkin, G. and Coult, N. (1998) A multiresolution strategy for reduction of elliptic pdes and eigenvalue problems. *Appl. Comp. Har. Anal.*, **5**,129–155.

[20] Hackbusch, W. (1985) *Multigrid Methods and Applications.* Springer-Verlag, New York.

[21] Knapek, S. (1998) Matrix-dependent multigrid homogenization for diffusion problems. *Siam J. Sci. Comput.*, **20**(2),515–533.

[22] Coifman, R.R. and Lafon, S. (2004) Geometric harmonics. *Appl. Comp. Har. Anal.*, in press.

[23] Fowlkes, C., Belongie, S., Chung, F., and Malik, J (2004) Spectral grouping using the nyström method. *IEEE PAMI*, **26**(2), 214–225.

[24] Williams, C. and Seeger,M (2001) Using the Nyström method to speed up kernel machines. In TK Leen, TG Dietterich, and V Tresp, editors, *Advances in neural Information Processing Systems 13:Proceedings of the 2000 Conference*, 682–688.

[25] Slepian, D. and Pollack, H.O. (1961) Prolate spheroidal wave functions, fourier analysis and uncertainty i. *The Bell System technical journal*, **40**, 43–64.

[26] Slepian, D (1964) Prolate spheroidal wave functions, Fourier analysis and uncertainty iv: extensions to many dimensions;generalized

prolate spheroidal wave functions. *The Bell System technical journal*, **43**, 3009–3058.

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Good Continuation in Layers:
Shading flows, color flows, surfaces and shadows

Ohad Ben-Shahar
Computer Science
Ben Gurion University
Israel
ben-shahar@cs.bgu.ac.il

Andreas Glaser
Applied Mathematics
Yale University
New Haven, CT 06520-8285
andreas.glaser@yale.edu

Steven W. Zucker
Computer Science
Yale University
New Haven, CT 06520-8285
steven.zucker@yale.edu

Abstract

We extend the concept of good continuation in a uniform fashion from boundaries to shading, hue, and texture. Each has the property that local measurements yield an orientation, which we explicitly establish for hue using geometric harmonic techniques. Good continuation arises in a geometric sense, because these orientations all vary smoothly in an appropriate sense. Thus they correspond to flows. Taken together they define a layered set of flows, in the sense the "horizontal" computations within each flow provide global consistency while "vertical" computations across flows enable the identification of shading and shadowing and different types of edges. Evidence is reviewed that primate visual systems enjoy such an organization.[1]

> "...space and color are not distinct elements but, rather, are interdependent aspects of a unitary process of perceptual organization." Kanizsa [17]

1. Introduction

Image segmentation is normally taken to be that process of partitioning the image into a complete cover of non-overlapping regions, with the boundaries of these regions related to the (projected) boundaries of objects in the world. One source of complexity in this process is shadowing, by which image intensities vary both as a function of surface orientation (e.g., shading) and as a function of light sources (e.g., cast shadows). Land's *retinex theory* [19] suggested one way to manage this complexity, by ascribing abrupt image changes to material (or reflectance) discontinuities and smooth gradient changes to lighting. This developed into the intrinsic image concept [30], which emphasized that surface properties, geometry, and lighting all map into the image, and suggested representing them separately as images. Undoing this map clearly involves an inverse problem, which requires a model of some sort. One possibility is to try to learn the context of every possible measurement, a type of pseudoinverse [28]. Here we extend the notion of context in a different way, by considering natural images such as those in Fig. 1. Notice how space,reflectance, and lighting conspire together. We seek to find a representation rich enough to support unwinding this.

The first requirement for such a representation is that it be rich enough to capture the above phenomena. But unlike special purpose algorithms applicable in one situation (e.g., [16, 13]), our second requirement is that it be general purpose. That is, the information that it makes explicit must support computations for unraveling many such phenomena.

We do not yet have a formal solution to this problem that we can prove is complete. Instead, and consistent with the goals of this Workshop, we develop an argument based on a neurobiological analogy, several steps of which have been formalized and are complete. The demonstrations in the final section of this paper involve phenomena beyond the current capability of any single existing algorithm, and provide counterexamples to many. Constructively, however, we submit that any final solution will have an intermediate representation at least as rich as the one we describe. Thus we see the contribution of this Workshop submission as consisting of (i) an enlargement of the framework for perceptual organization informed by (ii) the rich foundation for perceptual organization in primate visual systems.

The core of our argument is that good continuation applies to several key domains: boundaries, intensity (shading); hue; texture; saturation, and so on, all of which enjoy a certain differential geometric structure. It is this structure that relates to the Gestalt notion of *good continuation*. Computationally we propose a layered representation—similar in spirit to intrinsic images [30]—but different in

[1] Acknowledgements: Research supported by AFOSR, DARPA, ONR and the Toman and Frankel Funds from Ben-Gurion University.

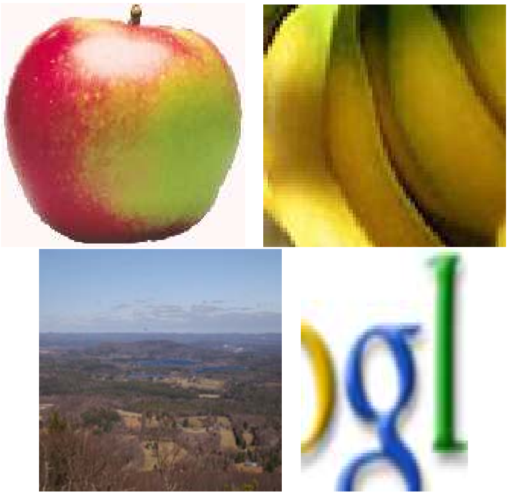


Figure 1. The rich interaction between surfaces, lighting, pigmentation, and atmosphere work together to provide a diversity of appearance phenomena in natural images. To simply claim that "apples are red" or "bananas are yellow" or "the sky is blue" amounts to an assumption that physical processes in the world are constant in a way that only artificial examples can really achieve.

that all share the property that they are flows in a technical sense. This is what we meant by layered flows implied in the title, and computations across these flows then reflect subtle lighting, surface, and space interactions.

Fig. 1 illustrates this point in several different domains (see also [3]. Apples are not a single color; rather, fruits mature differentially and this is reflected in their pigmentation. Attempts to remove these slow variations as lighting are one reason why lightness and color constancy algorithms have problems. Atmospheric depth effects impose a blue tint with distance because of increased scattering and in spite of surface reflection effects. Mutual illumination and color bleeding mix everything.

We approach the *lift* of these images into layered flows in two stages, both of which are mathematical but motivated by biology. We concentrate on one flow (from the color pathway) because, as will become clear below, the others fit naturally into our framework and are more widely discussed in the literature. Specifically, we first consider the question of how to represent color information as a dimensionality-reduction problem, which leads formally to intensity-hue-saturation coordinates at each point. This is important for us, because it suggests that there is more to color processing than simple detection tasks (consider: locate a red fruit among green foliage [27]) for which the standard cone pigments are tuned. We next consider (hue) interactions between points and adopt a technique previously used to denoise color patterns to articulate the flow of hue across image coordinates. The resultant computations are then run on the examples in Fig. 1.

2. Representation of Color at a Point

Take as data the Munsell patches considered as points in wavelength space. While wavelength-space is rather high-dimensional, our strategy is motivated by the observation that colors are not randomly distributed thoughout wavelength space, but rather occupy only a small portion of it. One possibility, suggested by the visual photopigments in primates, is that this structured space of colors is 3-dimensional. While this is a classical view of color, many of the classical algorithms have been modified in an *ad hoc* fashion to take account of non-linearities among colors (e.g., Multi-Dimensional Scaling). For this reason we use a new algorithm ([10, 11]) derived from the *geometric harmonics* (reviewed below) that can handle inherently non-linear data. It is in the class of spectral methods, and is related to [4].

2.1. Geometric Harmonics

Let $X = \{x_1, x_2, ..., x_N\}$ be the set of data points, in this case Munsell patches, with each $x_i \in R^n$. We seek to find a projection of these data into much lower dimension, under the assumption that they are not randomly distributed thoughout R^n but rather that they lie on (or near) a lower-dimensional manifold embedded in R^n.

The structure of the data are revealed via a symmetric, positivity-preserving, and positive semi-definite *kernel* $k(x, y)$, which provides a measure of similarity between data points. The result is a graph, with edges between nearby (according to the similarity kernel) data points. (The similarity value can be truncated to 0 for all but very similiar points.)

From this we construct a diffusion kernel $a(x, y)$ on the data set using the weighted graph Laplacian normalized as follows:

$$a(x, y) = \frac{k(x, y)}{\nu(x)}, \tag{1}$$

where $\nu = \sum_{y \in X} k(x, y)$. Note that, although symmetry is lost, we do have $\sum_{y \in X} a(x, y) = 1$ so the kernel $a(x, y)$ can be interpreted as the transition matrix of a Markov chain on the data X. The kernel $a^{(m)}$ of the m^{th} power of this matrix then represents the probability of getting from x to y in m steps.

If we now define the averaging operator for a function f defined on the data:

$$Af(x) = \sum_{y \in X} a(x, y) f(y) \tag{2}$$

then A admits a spectral theory. To develop this we symmetrize a by:

$$\tilde{a}(x, y) = \frac{\sqrt{\nu(x)}}{\sqrt{\nu(y)}} a(x, y) \tag{3}$$

which makes $\tilde{a}$ symmetric and positive semi-definite (although no longer row-stochastic). The spectral decomposition is then given by $\tilde{a} = \sum_{i \geq 0} \lambda_i^2 \phi_j(x)\phi_j(y)$ with the important consequence

$$\widetilde{a^{(m)}}(x,y) = \sum_{i \geq 0} \lambda_i^{2m} \phi_j(x)\phi_j(y) \tag{4}$$

where $\lambda_0 = 1$.

Increasing powers of the operator A can be obtained by running the chain through the spectral decomposition. This gives rise to the family of *diffusion maps* $\{\Phi_m\}_{m \in N}$ given by

$$\Phi_m(x) = \begin{pmatrix} \lambda_0^m \phi_0(x) \\ \lambda_1^{2m} \phi_1(x) \\ \vdots \end{pmatrix} \tag{5}$$

Diffusion distances $D_m^2(x,y) = \tilde{a}^{(m)}(x,x) + \tilde{a}^{(m)}(y,y) - 2\tilde{a}^{(m)}(x,y)$ within the high-dimensional measurement space then approximate Euclidean distance in the diffusion map space.

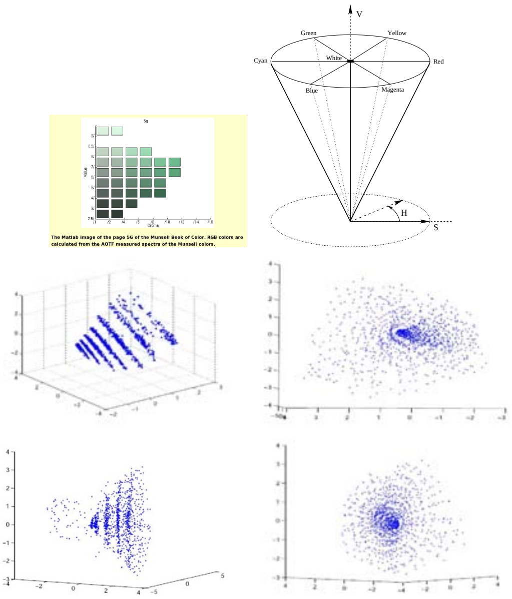
2.2. The Munsell Color Space

The Munsell [22] patches were chosen according to human psychophysics, with each step between patches perceptually equal, and they are now known to be physiologically relevant [31, 29, 15]. Thus they represent data spanning those portions of color space relevant to our interactions with the visible world. We now seek to understand whether these data lie on or near a well-defined structure in wavelength-space.

Two experiments were performed. We used $N = 1269$ patches, each with $n = 421$ wavelengths (380nm - 800nm in 1nm steps). The kernel is $\exp(-d_{ij}^2/\sigma)$ where d_{ij} is the Euclidian distance between patch i and patch j. While the patch data are given in no particular order, the geometric harmonic map arranges them so that patches are close to one another provided the diffusion distance between them in wavelength space is small. The results are shown in Fig. 2. Note that the natural representation emerges—intensity, hue, saturation—even though the hue (color circle) is non-linear. The diffusion maps recover the Munsell representation, thus demonstrating that the structure is in the wavelength data. In the second experiment we first projected the wavelength data through the human cone photopigments; and again the color circle emerged (Fig. 2, bottom).

3. Spatio-spectral Interactions

Now that we know there is a preferred representation for color at a point, we next consider the question of how colors interact between nearby points. We first observe that the primate visual system is well organized to address this problem. While it is widely accepted that perceptual organization is first accomplished via the long-range horizontal connections in superficial V1, consideration of these connections has been limited to orientation good continuation for boundaries ([24, 1, 2]) and textures ([7]). However, there exists a specialized structure for color (and contrast) information in the cytochrome oxidase blobs, within which neurons also enjoy long-range horizontal interactions (Fig. 3 [32]). We submit that it is precisely these connections that implement a geometry for hue (and color) that is formally analagous to that for texture[7] and shading [9, 21] flows. A sketch of this geometry is developed next. The extension to include boundaries is in [5].



The Matlab image of the page 5G of the Munsell Book of Color. RGB colors are calculated from the AOTF measured spectra of the Munsell colors.

Figure 2. Geometric harmonics organize Munsell color patches. (**top row, left**) Typical "page" of the patch data used in the experiment. Data from `http://spectral.joensuu.fi-/databases/download/munsell_spec_matt.htm`. (**right**) Classical intensity, hue, saturation color space. Note that hue is organized around the circle. (**middle row**) The geometric harmonic organization of the Munsell data. Each point represents a single patch, and the scatterplots show the distribution of points in the subspace spanned by the first three non-trivial eigenfunctions. Two views are shown, with (**left**) illustrating different clusters according to the Munsell chromaticity parameters and (**right**) a view showing the hue circle. That this non-linear organization of the data is recovered by geometric harmonics is significant because it provides the foundation for the next, geometric stage of processing. (**bottom row**) Organization of the Munsell data first projected through the three human cone photopigments. Since the two views are essentially the same as (**middle**), the Munsell representation is largely invariant to the order of projection.

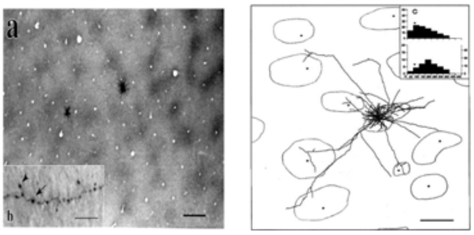


Figure 3. The cytochrome oxidase blobs in superficial primate visual cortex are specialized for the processing of color. The (**left**) figure shows the blobs selectively stained to highlight their locations regularly interspersed between orientation hypercolumns. (**right**) When single cells are filled with dye, their long-range connections become clear. Note how axons tend to terminate within (or near) other cytochrome oxidase blobs (drawn in outline). We submit that it is these long-range connections that enforce "good continuation" between hues at nearby positions. Images courtesy of E. Callaway, Salk Insitute.

3.1. Geometry of Hue Fields

Within the (intensity, hue, saturation) color space, the hue component across the image is a mapping $\mathcal{H} : \mathbb{R}^2 \rightarrow \mathcal{S}^1$ and thus can be represented as a unit length vector field over the image. In many images this hue field is piecewise smooth (Fig. 4) with singularities corresponding to significant scene events (e.g., occlusion boundaries or material changes).

The frame field [23] obtained by attaching a (tangent, normal) frame $\{E_T, E_N\}$ to each point in the image domain is the representation suggested by modern differential geometry. This provides a local coordinate system in which the hue vector and related structures can be represented. Most importantly among these are the covariant derivatives of E_T and E_N, which represent the initial rate of change of the frame when it is moved in a direction v expressed by the connection equation [23]:

$$\begin{pmatrix} \nabla_V E_T \\ \nabla_V E_N \end{pmatrix} = \begin{bmatrix} 0 & w_{12}(V) \\ -w_{12}(V) & 0 \end{bmatrix} \begin{pmatrix} E_T \\ E_N \end{pmatrix} \quad (6)$$

The coefficient $w_{12}(V)$ is a function of the tangent vector V, which represents the fact that the local behavior of the flow depends on the direction along which it is measured. $w_{12}(V)$ is a linear 1-form, so it can be represented with two scalars at each point:

$$\begin{aligned} \kappa_T &\stackrel{\triangle}{=} w_{12}(E_T) \\ \kappa_N &\stackrel{\triangle}{=} w_{12}(E_N) \ . \end{aligned} \quad (7)$$

We call κ_T the hue's *tangential curvature* and κ_N the hue's *normal curvature* - they represent the rate of change of the hue in the tangential and normal directions, respectively.

Since the local behavior of the hue is characterized (up to Euclidean transformation) by a pair of curvatures, it is natural to conclude that nearby measurements of hue should relate to each other based on these curvatures. Put differently, measuring a particular curvature pair $(\kappa_T(q), \kappa_N(q))$ at a point q should induce a field of coherent measurements, i.e., a hue function $H\tilde{U}E(x,y)$, in the neighborhood of q. Coherence of $HUE(q)$ to its spatial context $HUE(x,y)$ can then be determined by examining how well $HUE(x,y)$ fits $H\tilde{U}E(x,y)$ around q. Clearly, this should be a function of the local hue curvatures $(\kappa_T(q), \kappa_N(q))$, it should agree with these curvatures at q, and it should extend around q according to some variation in both curvatures

While many local coherence models $H\tilde{U}E(x,y)$ are possible, we exploit the fact that the hue field is a unit length vector field which suggests that it behaves similarly to oriented texture flows [6, 7] and adopt a similar curvature-tuned local model.

$$H\tilde{U}E(x,y) = tan^{-1}\left(\frac{\kappa_T(q)x + \kappa_N(q)y}{1 + \kappa_N(q)x - \kappa_T(q)y}\right) \ . \quad (8)$$

Unlike texture flows, however, the local model for the hue function is not a *double* helicoid since the hue function takes values in $[\pi, \pi)$ where texture flows are constrained to $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right)$.

This local model possesses many properties that suit good continuation; in particular it is both a minimal surface in the $(x, y, H\tilde{U}E(x,y))$ representation and a critical point of the p-harmonic energy for all p. It is also the only local model that does not bias the changes in one hue curvature relative to the other, i.e., it satisfies

$$\frac{\kappa_T(x,y)}{\kappa_N(x,y)} = \text{const} = \frac{\kappa_T(q)}{\kappa_N(q)} \ .$$

Examples of the model for different curvature tuning is illustrated in Fig 5. A detailed technical account of the model in the texture flow domain can be found in [7].

4. Examples of Flows

We now illustrate the above computations on several examples. We begin with artificial ones, to illustrate the points most clearly, then proceed to natural ones to illustrate the complexities that arise.

We stress that, for space reasons, some of these flows are not visible unless one zooms in to enlarge the manuscript.

In the first Fig. 6, we show one of the few examples from the psychophysical literature. In an important paper, Kingdom [18] created images consisting of superimposed sinusoids, one in brightness and the other in color. He demonstrated that it is the intensity component that drives the impression of shape-from-shading, while the color information appears "painted" onto the undulating surface. We reproduced this separation with our flows, from which it follows that the shading flow is sufficent (for these examples) to derive the shape.

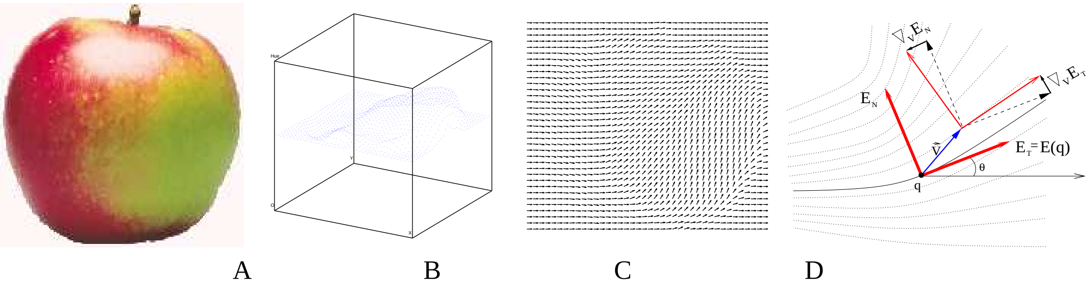


Figure 4. Color images of natural objects are piecewise smooth and the hue flow captures this. **(A)** An apple with varying hue. **(B)** A representation of hue as a scalar field, with value corresponding to height. **(C)** The hue field, with each value represented as a vector pointing to location on the hue circle. **(D)** The geometry of the hue flow, illustrating that nearby values can be represented as a differentiable frame field that is tangent (and normal) to the streamlines of the flow. Interations between nearby hue values then correspond to an (infinitesimal) transport of the frame in direction V, which rotates it according to the connection form of the frame field. Since E_T, E_N are unit length, their covariant derivative lies in a normal direction, regardless of V. This diagram also suggests a relationship between hue and texture and shading flows.

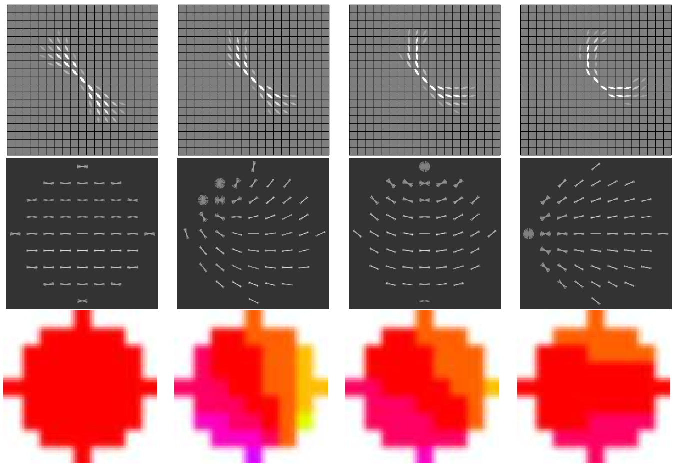


Figure 5. Illustration of the different types of compatibility fields that can be used for early forms of good continuation. In each case the central unit is supported by the contextual arrangement of surrounding units, and can be used as the constraints within quadratic programming, relaxation labeling, and belief propagation engines. **(top)** For boundary continuation, the orientation at a position is enhanced by consistent tangential (co-circular) boundary measurements at nearby positions [24, 14] **(middle)** For oriented texture measurements, both tangential and normal curvatures arise. Similar models can be used for shading flows, which are the tangent fields to the intensity level sets [8]. **(bottom)** For hue flows the orientations are replaced by colors. In the first column zero curvature continuations are shown. In the last column, a single large curvature is shown. For the texture and hue compatibilities, the tangential curvature is zero and the normal curvature is not. Note the emergence of singularities.

The shading flow is estimated by evaluating a gradient operator (an orientationally-selective receptive field tuned to low spatial frequency) over the image. It demonstrates one role for the long-range interactions: correcting local artifacts in shading flow estimation.

Our next examples (Fig. 7) on artificial images confirm the classical view that color remains invariant across shadows while shading effects surface percepts [25]. This is most clear in the plastic sphere, and the same effect is reproduced in the Google logo, which appears both 3-dimensional and colored. However, unlike the plastic sphere, there are no mutual illumination effects.

The next examples show how hue can vary over a natural object. Fig. 4 shows the hue flow for an apple, and Fig. 8 is a close-up of a woman's face in which a blush has been introduced. Note in particular how variant the "color" is, a point of some relevance to both face identification and emotional estimation. Hue can also vary systematically over a scene. Atmospheric depth scattering is shown in Fig. 9.

Our next two examples illustrate the beautiful complexity of shading, hue, and boundary interactions. The first shows an apple photographed on a highly reflective surface in bright sunlight (Fig. 10). The flows are varied with respect to one another and with respect to the boundaries (of both the apple and the shadow). In particular, the mutual illumination modulating the shadow [20] introduces a smooth shading flow not unlike the one for the plastic sphere or the Kingdom examples but this time due to a lighting effect and

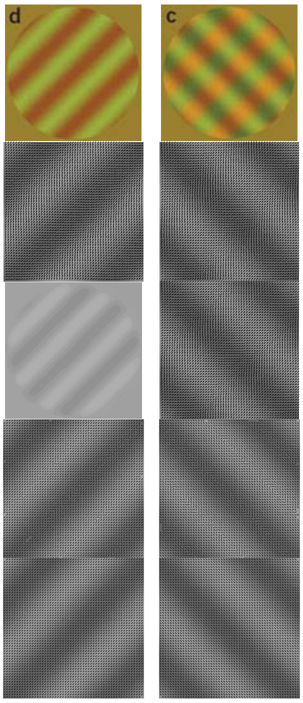


Figure 6. Results on the test Kingdom images. Note how both provide the impression of an undulating surface with color on it. The left column is Kingdom Fig. 2d; the right column is Kingdom Fig. 2c. From top to bottom are original images; initial estimate of shading flow (tangents to intensity level sets); final estimate of shading flow; initial estimate of hue flow; final estimate of hue flow. The shading flow corresponds to the undulations; the hue flows are smooth and do not interfere with them.

not a surface normal effect. The mutual illumination effect is also strong on the bananas image (Fig. 11), which also illustrates a shading flow effect due to a highly diffuse cast shadow. In this case the cast shadow phenomenon is readily identified, because the hue flow is constant across it.

Our final example (Fig. 12) illustrates the complement to shading and hue; notice how the hue remains invariant through the highlight, even though it is a complex pattern for the pepper.

5. Summary and Conclusions

Perceptual organization was viewed within Gestalt psychology as pervasive in perception, but discussion of such issues in computer vision is significantly more limited. Our goal in this paper was to take a step back and raise the profile of questions for which P.O. is relevant. Following a biological analogy, we introduced the construct of multiple (spatially) aligned flows within which Gestalt good continuation can be enforced geometrically but between which information can be inferred about the many complexities of lighting, space, and geometry. The computation of each flow was global, based on local measurements and differential (covariant derivative) constraints between them. At the same time the computation of each flow was local within an information (sometimes within a sensor) source, and logical relationships between flows provide a new foundation for

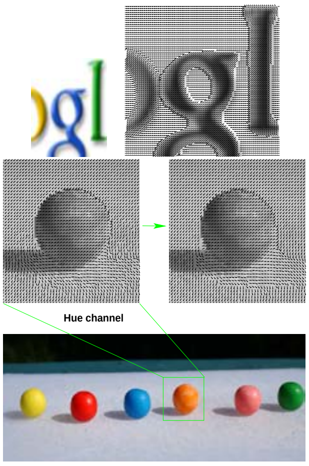


Figure 7. Shading and hue flows for artificial objects. Although the shading flow fields are not shown, notice how the hue flows (superimposed on the original image) are constant over the "plastic" objects. This is the way such materials were designed. The case of the sphere also introduces two more complex lighting effects. First, note how the hue flow remains constant through the shadow. This is a classical cue for separating shadow boundaries from surface boundaries. (Surface boundaries are taken to involve different materials, and therefore a hue discontinuity together with the intensity discontinuity.) Second, and less familiar, is the mutual illumination between the sphere and the tabletop, which is captured by the hue flow but not the shading flow. The left magnification shows the initial local measurements of hue; the right magnification shows the converged hue flow. A boundary has been introduced around the hue flow on the table top illustrating an elongation in the direction of the source.

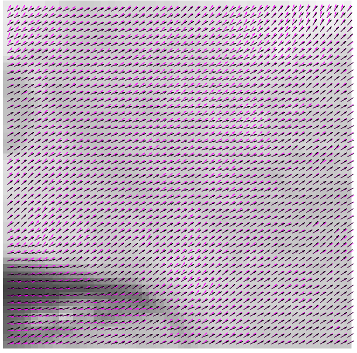


Figure 8. Hue flows vary for natural objects. This shows a portion of a woman's face (the lips are lower left) when she is blushing (blue vectors) and not blushing (black vectors). Note how hue varies both spatially and as a function of emotional and physical states.

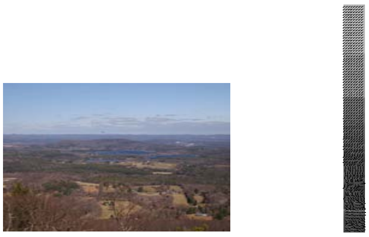


Figure 9. Hue flows and atmospheric depth effects. The flow is shown along a thin strip on the right side of the photograph. Note the dominant shift toward blue for the upper half.

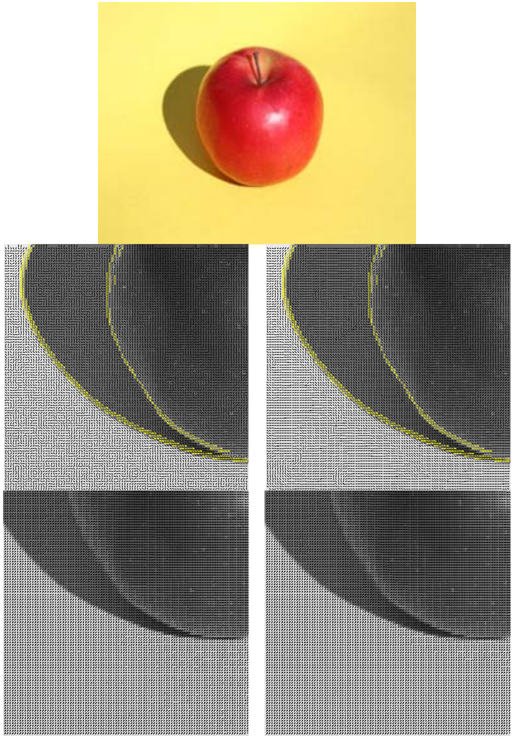


Figure 10. An image of an apple on colored cardboard in bright sunlight. It illustrates the complexities that can arise both for shading due to surface irregularities from packing and from mutual illumination. In particular, the shaded area now exhibits a shading flow derived from mutual illumination, in which the gradient decreases in magnitude away from the concavity between the apple and the table. At the same time, there is strong mutual illumination between the apple and the cardboard and the cardboard and the apple. The result are smooth shading and hue flows, with discontinuities at neither object nor shadow edges.

many computer vision computations. Hue flows smoothly through shadows, while intensity often jumps. Shading flows smoothly over many man-made objects, while hue is often constant. Natural objects often imply smooth shading and hue flows, although they are typically independent of one another. The involvement of boundaries is both necessary and complicated [12].

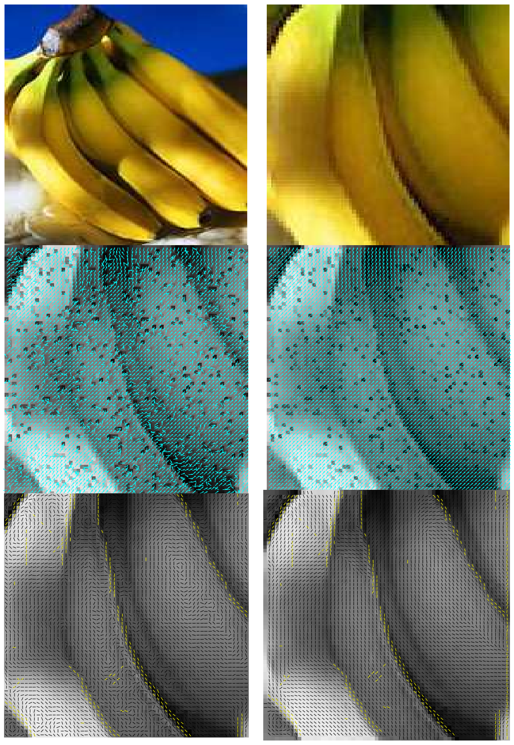


Figure 11. A photograph of bananas illustrates the richness of mutual illumination in a complex scene. The result is an essentially constant hue flow (middle row, left: initial measurement; right: consistent flow). The shading flow (**bottom**) illustrates a special interaction between boundaries and shading flows, in which multiple surface fold away from each other along them. Such situations are geometrically rare.

While the list of interactions must be extended (motion and stereo should at least be included), it is useful to conclude on an enlargement of the biological metaphor underlying this paper. The centrality of long-range horizontal connections as defining each flow suggests that the flows be layered on top of one another, enabling "vertical" connections for their interactions. Recent breakthoughs in color processing demonstrate that hue and orientation are not independent, as was once thought, and that such vertical connections exist [26]. Computationally it remains an open question whether only two interaction "dimensions" suffice.

References

[1] Y. Adini, D. Sagi, and M. Tsodyks. Excitatory-inhibitory network in the visual cortex: Psychophysical evidence. *Proc. Natl. Acad. Sci. U.S.A.*, 94:10426–10431, 1997. 3

[2] W. Beaudot and K. Mullen. How long range is contour integration in human color vision. *Visual Neurosci.*, 20:51–64, 2003. 3

[3] J. Beck. *Surface Color Perception*. Cornell University Press, 1972. 2

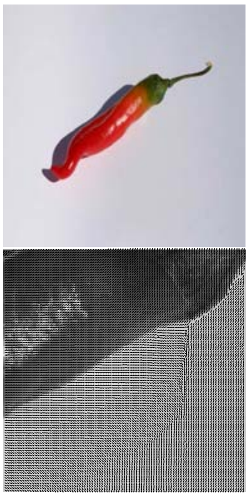


Figure 12. A photograph of a pepper illustrates how the hue flow remains constant through highlights, even when the hue is varying. The mutual illumination on the background is very interesting as well. (**top**) Pepper image. (**bottom**) Hue flow through a portion of the pepper image. The flow is superimposed on the image for identification purposes only.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6(15):1373–1396, June 2003. 2

[5] O. Ben-Shahar, P. Huggins, and S. Zucker. On computing visual flows with boundaries: The case of shading and edges. In *Workshop on Biologically Motivated Computer Vision*, 2002. 3

[6] O. Ben-Shahar and S. Zucker. On the perceptual organization of texture and shading flows: From a geometrical model to coherence computation. In *Proc. Computer Vision and Pattern Recognition*, pages 1048–1055, 2001. 4

[7] O. Ben-Shahar and S. Zucker. The perceptual organization of texture flows: A contextual inference approach. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(4):401–417, 2003. 3, 4

[8] O. Ben-Shahar and S. Zucker. Geometrical computations explain projection patterns of long range horizontal connections in visual cortex,. *Neural Comput.*, 16(3):445–476, 2004. 5

[9] P. Breton and S. Zucker. Shadows and shading flow fields. In *Proc. Computer Vision and Pattern Recognition*, pages 782–789, 1996. 3

[10] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc. Nat. Acad. Sci. (USA)*, 102(21):7426 – 7431, 2005. 2

[11] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Multiscale methods,. *Proc. Nat. Acad. Sci. (USA)*, 102(21):7432 – 7437, 2005. 2

[12] J. Elder and S. Zucker. Evidence for boundary-specific grouping. *Vision Res.*, 38(1):143–152, 1998. 7

[13] K. Garg and S. Nayar. When does a camera see rain? *ICCV*, 2005. 1

[14] W. Geisler, J. Perry, B. Super, and D. Gallogly. Edge co-occurrence in natural images predicts contour grouping performance. *Vision Res.*, 41(6):711–724, 2001. 5

[15] A. Hanazawa, H. Komatsu, and I. Murakami. Neural selectivity for hue and saturation of colour in the primary visual cortex of the monkey. *Eur. J. Neurosci.*, 12:1753–1763, 2000. 3

[16] B. Horn and M. Brooks, editors. *Shape from Shading*. MIT Press, Cambridge, MA, 1989. 1

[17] G. Kanizsa. *Organization in Vision: Essays on Gestalt Perception*. Praeger Publishers, 1979. 1

[18] F. Kingdom. Color brings relief to human vision. *Nature Neuroscience*, 6(6):641–644, 2003. 4

[19] E. Land and J. McCann. Lightness and retinex theory. *American Journal of Optical Society of America*, 61:1–11, 1971. 1

[20] M. Langer. When shadows become interreflections. *Int. J. Comput. Vision*, 34(2/3):193–204, 1999. 5

[21] S. Lehky and T. Sejnowski. Network model of shape-from-shading: neural function arises from both receptive and projective fields. *Nature*, 333:452–454, 1988. 3

[22] A. Munsell. *A Color Notation*. G.H.Ellis, Boston, 1905. 3

[23] B. O'Neill. *Elementary Differential Geometry*. Academic Press, 1966. 4

[24] P. Parent and S. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(8):823–839, 1989. 3, 5

[25] M. Ruzon and C. Tomasi. Color edge detection with the compass operator. In *Proc. Computer Vision and Pattern Recognition*, pages 160–166, 1999. 5

[26] R. Shapley and M. Hawken. Neural mechanisms for color perception in the primary visual cortex. *Curr. Opin. Neurobiol.*, 12:426–432, 2002. 7

[27] P. Sumner and J. Mollon. Chromaticity as a signal of ripeness in fruits taken by primates. *Journal of Experimental Biology*, 203(13):1987–2000, 2000. 2

[28] M. Tappen, W. Freeman, and E. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(9):1459–1472, 2005. 1

[29] T. Wachtler, T. Sejnowski, and T. Albright. Representation of color stimuli in awake macaque primary visual cortex. *Neuron*, 37:681–691, 2003. 3

[30] A. Witkin and J. Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 481–542. Academic Press, 1983. 1

[31] Y. Xiao, Y. Wang, and D. Felleman. A spatially organized representation a colour in macaque cortical area v2. *Nature*, 421:535–539, 2003. 3

[32] N. Yabuta and E. Callaway. Cytochrome oxidase blobs and intrinsic horizontal connections of layer 2/3 pyramidal neurons in primate v1. *Visual Neurosci.*, 15:1007–1027, 1998. 3

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Geometries of sensor outputs, inference and information processing

Ronald R Coifman[1,2], Stephane Lafon[3], Mauro Maggioni[1,2], Yosi Keller[1,2], Arthur D Szlam[1,2], Frederick J Warner[1,2], Steven W Zucker[2,4]

[1]Department of Mathematics, [2]Program in Applied Mathematics, [4]Department of Computer Science, Yale University, 10 Hillhouse Ave, New Haven, CT, 06520; [3]Google Inc., 1600 Amphitheatre Pkw., Mountain View, CA, 94043.

ABSTRACT

We describe signal processing tools to extract structure and information from arbitrary digital data sets. In particular heterogeneous multi-sensor measurements which involve corrupt data, either noisy or with missing entries present formidable challenges. We sketch methodologies for using the network of inferences and similarities between the data points to create robust nonlinear estimators for missing or noisy entries. These methods enable coherent fusion of data from a multiplicity of sources, generalizing signal processing to a non linear setting. Since they provide empirical data models they could also potentially extend analog to digital conversion schemes like "sigma delta".

Keywords: Markov processes, multiscale analysis, diffusion on manifolds, Laplace-Beltrami operator.

1. FEATURE BASED FILTERING, DIFFUSIONS AND SIGNAL PROCESSING ON GRAPHS

A simple way to understand the effect of introducing similarity based diffusions on data[1–6] is provided by considering a regular gray level image in which we associate with each pixel p a vector $\nu(p)$ of features.[7,8] For example, a multi-band electromagnetic spectrum or the 5×5 sub-image centered at the pixel, or any combination of features. Define a Markov filter

$$A_{p,q} = \frac{\exp -\frac{||\nu(p)-\nu(q)||^2}{\epsilon}}{\sum_q \exp \frac{-||\nu(p)-\nu(q)||^2}{\epsilon}}\,, \tag{1}$$

where $\epsilon > 0$ is a small parameter comparable to the smallest distances between two feature vectors $\nu(p)$ and $\nu(q)$. Clearly the map ν is a bijection between pixels in the image and patches (or features). In particular every function on the pixels, such as the original image I itself, is also a function on the set of patches. With this identification, one can let the Markov filter $A_{p,q}$ act on an image.

The image I in figure 1 was filtered using the (nonlinear in the features) procedure described above where the feature vector $\nu(p)$ is the 5×5 patch around a pixel p:

$$I(p) = \sum_q A_{p,q} I(q) = \sum_q \frac{\exp -\frac{||\nu(p)-\nu(q)||^2}{\epsilon}}{\sum_q \exp \frac{-||\nu(p)-\nu(q)||^2}{\epsilon}} I(q)\,. \tag{2}$$

Observe that the edges are well preserved as patches translated parallel to an edge are similar and contribute more to the averaging procedure.[7,8] We should also observe that if we were to repeat the procedure on the filtered image we would get a numerical implementation of various nonlinear heat diffusions for image processing similar to those in PDE methods, such as those by Osher and Rudin.

It is useful to replace A by a bi-Markovian version of the form

$$A_{p,q} = \frac{\exp \frac{-||\nu(p)-\nu(q)||^2}{\epsilon}}{\omega(p)\omega(q)}$$

Send correspondence to R.R. Coifman. E-mail: coifman@fmah.com, Telephone: 1 203 432 1213vited Paper

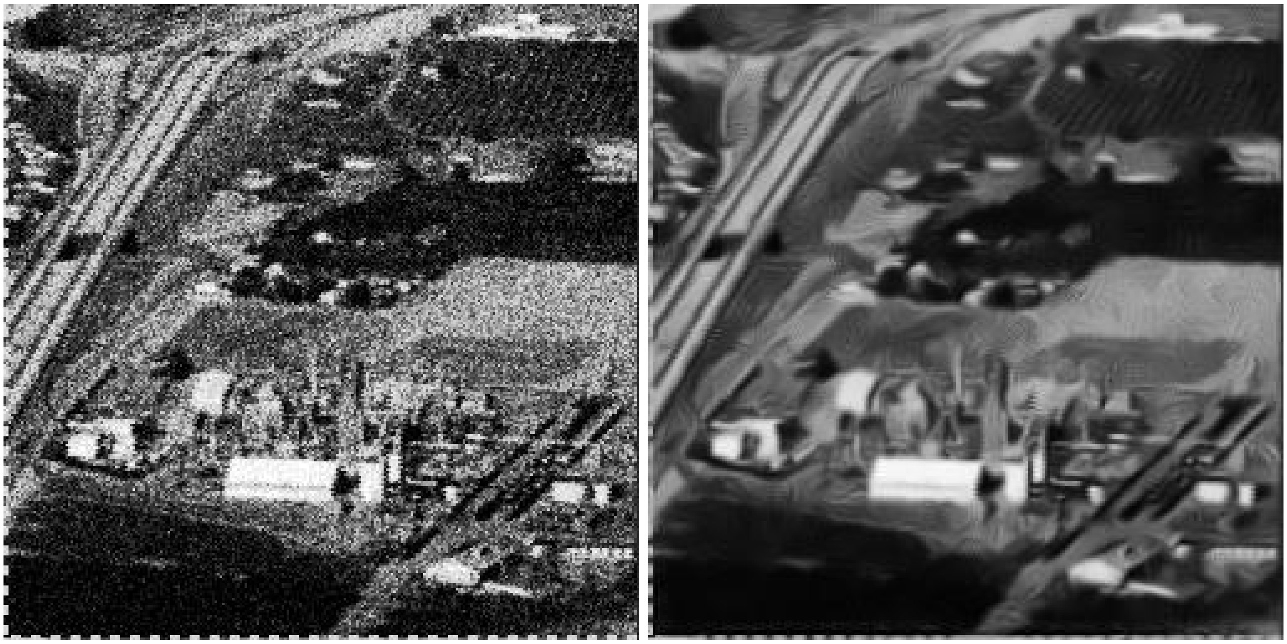


Figure 1. Left: original noisy image. Right: image denoised by application of the Markov matrix as in (1)

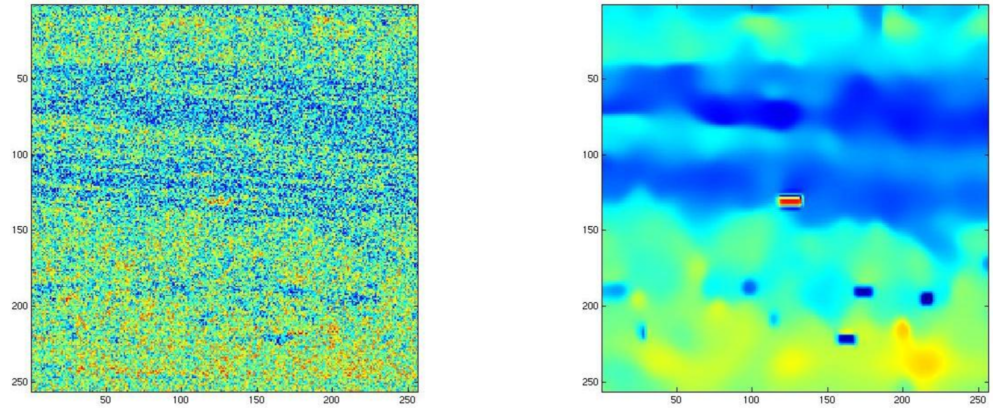


Figure 2. Left: original noisy image. Right: image denoised by application of the Markov matrix as in (1), but where features are local variances rather than pixel values in a patch around each pixel.

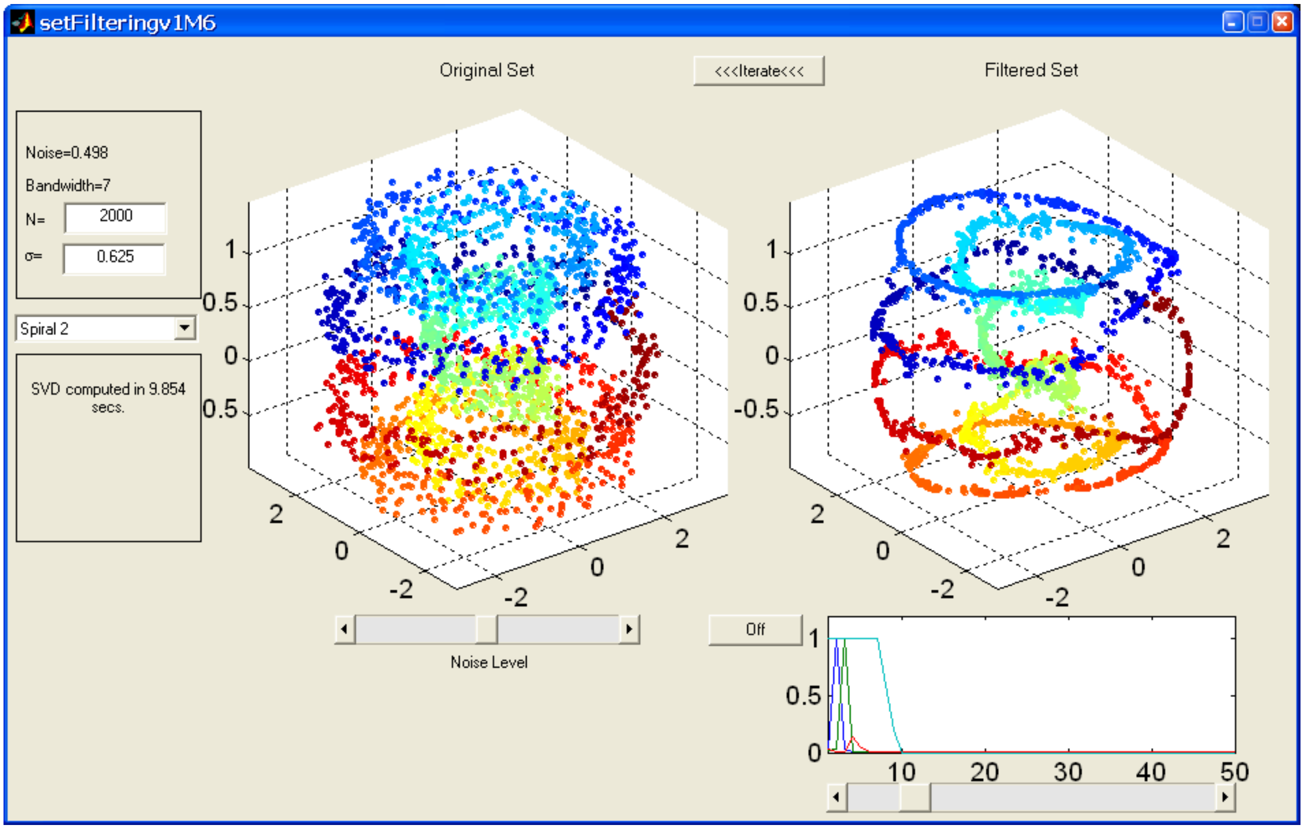
where the weights $\omega(\cdot)$ are selected so that A is Markov in p and q.

The noisy IR image in Figure 2 was filtered by N. Coult using a vector of 25 statistical features associated with each pixel.

The Markov matrix used for filtering, defines a diffusion on the graph of patches or features viewed as a subset of 25 dimensional Euclidean space. The eigenvectors of this diffusion permit us to compute all of its powers and to define a diffusion geometry and signal processing on this "image graph".[7]

For the next example consider 3 noisy sensors measuring the x, y, z coordinates of a trajectory in three dimensions. We could try to denoise each coordinate separately. Or use the position vector as as a feature vector as we did for the images above. See Figure 1.

The construction above should be viewed as signal processing on the data graph. We view all points of the trajectory as a data graph,ie data points p and q are vertices and $A_{p,q}$ is the weight of the edge connecting them



Noise=0.498

Bandwidth=7

N= 2000

σ= 0.625

Spiral 2

SVD computed in 9.854 secs.

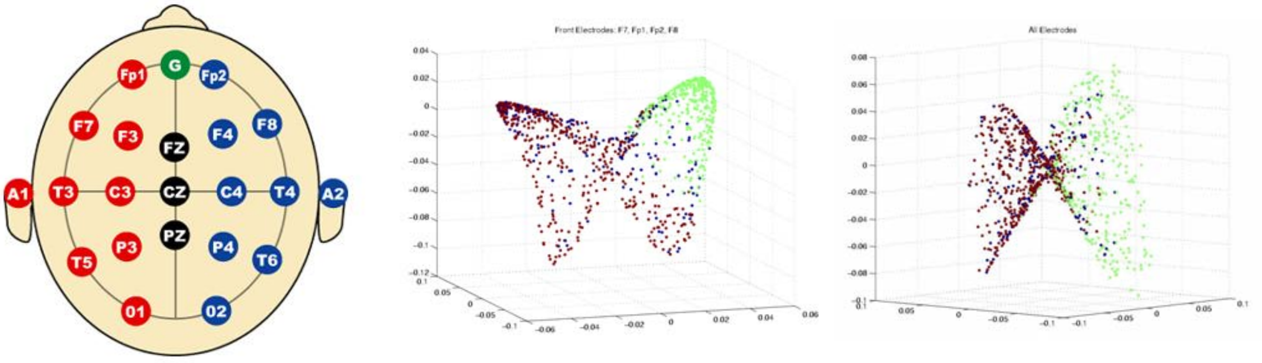


Figure 4. Left: standard position of electrodes in EEG. Middle: diffusion map of the responses to 4 electrodes, showing the nonlinear correlations and manifold-like structure of these responses. Right: diffusion map of the responses to all electrodes, exhibiting similar nonlinear correlations. In fact, the manifold structure obtained from measuring from all electrodes is very close to that obtained from 4 electrodes, suggesting that exploiting the nonlinear correlations would allow to use only 4 electrodes.

and define the diffusion map $\Phi_m^{(t)}$ at time t into m dimensional Euclidean space by

$$X_p \mapsto \Phi_m^{(t)}(X_p) := (\lambda_1^t \varphi_1(X_p), \lambda_2^t(X_p), \ldots, \lambda_m^t \varphi_m(X_p)) \qquad (3)$$

For a given t we determine m so that λ_{m+1}^t is negligible. The diffusion distance[1] at time t between $X_p^{(t)}$ and $X_q^{(t)}$ is given as

$$d_t^2(p,q) = A_{p,p} + A_{q,q} - 2A_{p,q} = \sum_l \lambda_l^{2t}(\varphi_l(X_p) - \varphi_l(X_q))^2 = ||\Phi_m^{(t)}(X_p) - \Phi_m^{(t)}(X_q)||^2 \,.$$

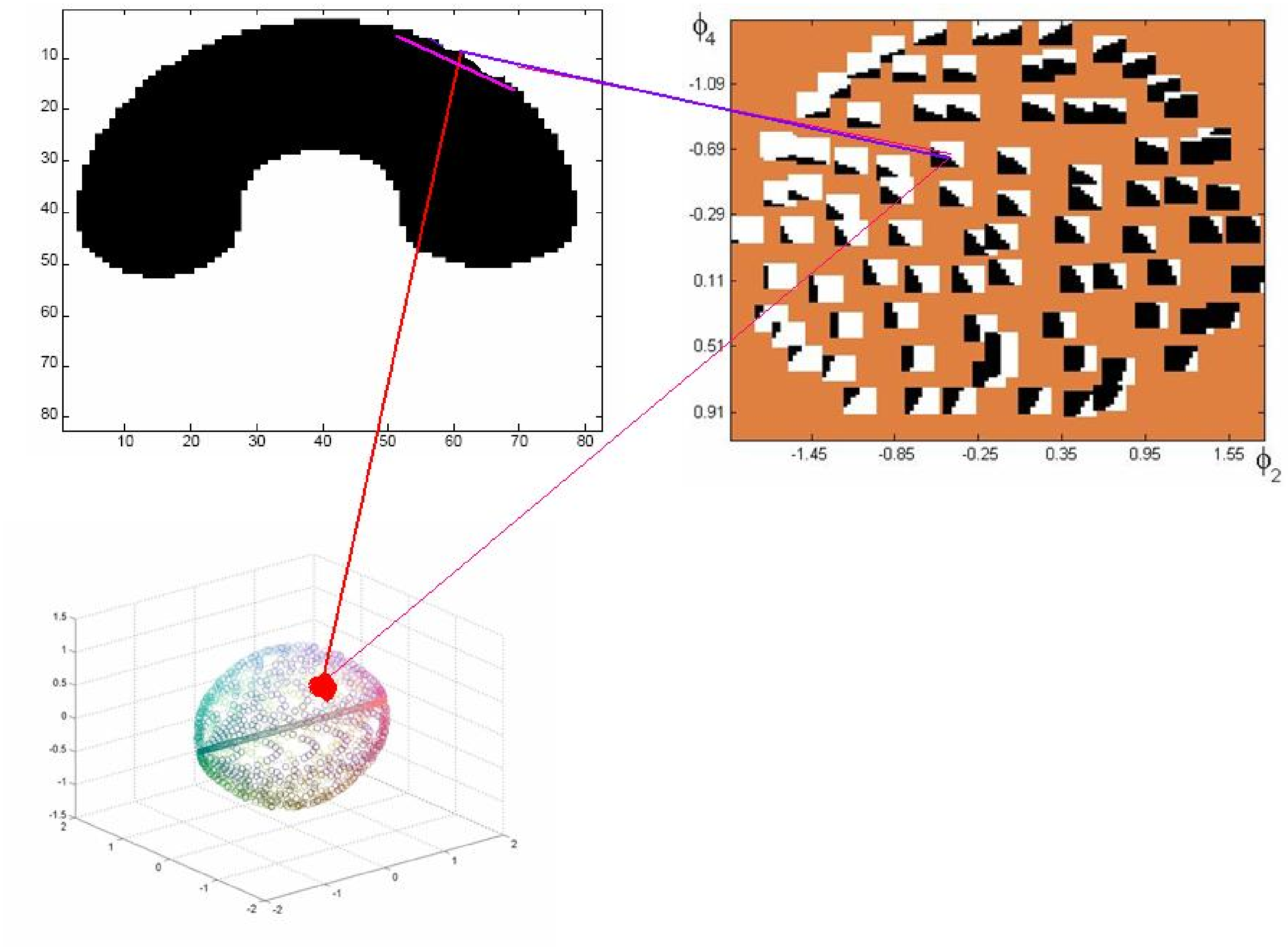
This map enables us to represent geometrically an abstract set of measurements on a sensor array (measurement space) as we illustrate on the following EEG example.[11]

The 20 electrodes measure coherent electrical activity in the brain. Mapping the configuration space of the measurements of 4 electrodes leads to the same configuration as for all 20. In the linear case this will be obtained by de-correlating the outputs , here however different locations of sources result in a different attenuation vectors ,or linear de-correlations. Here the first three nontrivial eigenvectors are used to map the data to three dimensions (diffusion map), see Figure 4. The implications are obvious 4 electrodes suffice to get essentially the same measurements , redundancy is useful to obtain a clean version.[11]

3. MULTISCALE STRUCTURES AND THE EMERGENCE OF ABSTRACT SENSOR FEATURES

It is possible to build a multiscale decomposition of a data graph simply by organizing the data into affinity folders where the affinity is measured through the diffusion distance at different time scales A simple algorithm[9] is obtained as follows Let x_j^{l+1} be a maximal sub-collection of points in $\{x_j^1\}$ (key-points at scale 1) such that $d_{t_l}(x_j^{l+1}, x_i^{l+1}) \geq \frac{1}{2}$, where x_j^0 are the original points, and $t_l = a2^l$, $l = 0, 1, 2, \ldots$. Then clearly each point is at distance less than a half at scale l from one of the selected key-points allowing us to create a folder labeled by the key-point. It is easy to modify to obtain a tree of disjoint folders by viewing each key point as the folder of points nearest to it, and reinterpret the distance as distance between folders.

When applied to text documents (equipped with semantic coordinates), this construction builds an automatic folder structure with corresponding keywords characterizing the folders.[4,7] While for text documents folders are just collection of related documents, and abstractions are collection of words in a given class, the situation is


10
20
30
40
50
60
70
80
10
20
30
40
50
60
70
80
ϕ_4
-1.09
-0.69
-0.29
0.11
0.51
0.91
-1.45
-0.85
-0.25
0.35
0.95
1.55
ϕ_2
1.5
1
0.5
0
-0.5
-1
-1.5
2
1
0
-1
-2
-2
-1
0
1
2

	"0"	**"1"**	**"2"**	**"3"**	**"4"**	**"5"**	**"6"**	**"7"**	**"8"**	**"9"**
zero	**0.90**	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.00	0.00
one	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
two	0.00	0.00	**0.96**	0.01	0.02	0.00	0.00	0.00	0.00	0.00
three	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
four	0.00	0.00	0.00	0.04	**0.96**	0.00	0.00	0.00	0.00	0.00
five	0.00	0.00	0.00	0.00	0.00	**0.97**	0.00	0.00	0.02	0.01
six	0.06	0.00	0.00	0.00	0.00	0.00	**0.90**	0.04	0.00	0.00
seven	0.03	0.00	0.00	0.00	0.00	0.00	0.03	**0.93**	0.00	0.00
eight	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	**0.95**	0.03
nine	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.02	**0.96**

where φ_α is a (e.g. wavelet) basis on Q, and $\varphi_\beta(r)$ is a (wavelet) basis on R. In the formula above

$$\delta_{\alpha,\beta} = \sum_{q,r} d(q,r)\varphi_\alpha(q)\varphi_\beta(r)\,,$$

where we accept this sum (as validated) only if various randomized averages using subsamples of our data lead to the same value of $\delta_{\alpha,\beta}$. In the calculation of D we only use accepted estimates for $\delta_{\alpha,\beta}$.

The wavelet basis can of course be replaced by tensor products of scaling functions or any other approximation method in the tensor product space, including other pairs of bases, one for q the other for r, including graph Laplacian eigenfunctions (we observe in passing that the singular value decomposition is a particular case of this construction). A direct method for filtering d or estimating D without the need to build basis functions can be implemented as at the beginning of this paper.

Define a Markov matrix $A = a[(r,q),(r',q')]$ (corresponding to diffusion on $Q \times R$) as

$$a[(r,q),(r',q')] = \frac{\exp\left(\frac{||\nu(r)-\nu(r')||^2}{\epsilon} + \frac{||\mu(q)-\mu(q')||^2}{\delta}\right)}{\sum_{r,q}\exp\left(\frac{||\nu(r)-\nu(r')||^2}{\epsilon} + \frac{||\mu(q)-\mu(q')||^2}{\delta}\right)} \tag{5}$$

Where the vector $\nu(r)$ is response column vector corresponding to the column r, and $\mu(r)$ is a sensor row vector.

The parameters epsilon, delta are chosen after randomized validation as described above. We can have an alternate definition of D as follows.

$$D(r,q) = \sum_{r,q} a[(r,q),(r',q')]d(r,q)\,.$$

Observe that the distances occurring in the exponent can be replaced by any convenient notion of distance or dissimilarities, and that any polynomial in A can be used to obtain a better filtering operation on the raw data.

A new combined graph can also be formed by embedding the graph $Q \times R$ into Euclidean space ,say by the diffusion embedding , followed by an expansion of the data $d(q,r)$ on this new structure, or by filtering as above on the new structure.

5.1. Markov Decision Processes

In the papers[14, 15] the multiscale analysis construction of diffusion wavelets is applied to Markov Decision Processes. Informally, and in a simplified version, one or more agents explore a given *state space* S by taking actions in each state from a set of actions A, and collect different *rewards* R, that we assume, to simplify the presentation, to depend only on the location and not on the action. Suppose we can model the state space as a finite graph (S, E, W) (the uncountable or continuous case can be handled as well), with edges E and weights W, and that the agent(s) explore the state space randomly accordingly to the Markov process P^π, parametrized by a (*policy*) π, which maps each state to a probability distribution of actions for that state. The reward function R is a real-valued function on S. The expected long term sum of discounted rewards when the agent follows the policy π is a function V^π on S, called (state) *value function.* It satisfies the so-called Bellman equation $V^\pi = R + \gamma P^\pi V^\pi$, $\gamma \in (0,1]$ being the discount factor, and hence $V^\pi = (I - \gamma P^\pi)^{-1}R$. In terms of potential theory, $(I - P^\pi)^{-1}$ is the Green's function (or fundamental matrix) of the "Laplacian" $I - P^\pi$, and V^π is the potential generated by the "charge" R under the diffusion P^π. Suppose for simplicity that P^π is reversible: it is then similar to a symmetric matrix T^π that generates a Markov diffusion semigroup $\{(T^\pi)^t\}$. The diffusion multiscale analysis allows to efficiently compute $(P^\pi)^t(x,y)$ for arbitrary t, medium and large, for one or multiple agents; it allows to effectively approximate the value function V^π, which is often piecewise smooth, performing a very useful dimensionality reduction,[14] where *ad hoc* basis functions were previously constructed by hand and were only available in particularly simple geometries. Finally, it allows to solve Bellman's equation directly, to high precision, in an efficient way. In[15] this method is compared with classical direct methods (often unfeasible because of their computational complexity of $\mathcal{O}(|S|^3)$), and with optimized iterative solvers.

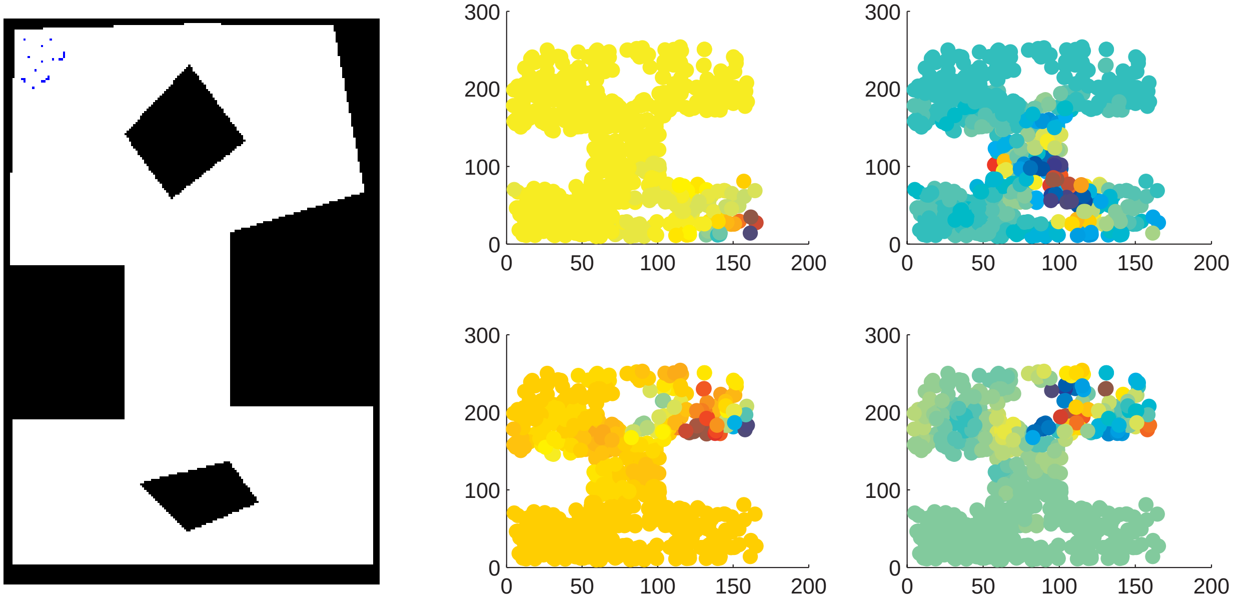


Figure 7. Left: continuous state space for a MDP, the actions are movements in the four cardinal directions, blue points represent positive rewards. Right: after a random exploration by the agent, multiscale bases functions are constructed on the state space: the color is proportional to the value of various scaling functions, which are automatically adapted to the state space. The value function can be projected onto this basis, in fact if the value function is piecewise smooth, only few elements of the basis (a number independent of the number of samples!) will be required to approximate the value function to a given precision.

6. CONCLUSIONS AND DISCUSSION

It is quite clear from the preceding descriptions that the data graph can be equipped with informative geometric structures which coherently integrate data and enable inference and interpolation. One of our main goals is to efficiently regress empirical functions on a data set, we have indicated various methods to build and approximate empirical functions, admitting natural extensions (generalization) off the known measured data. We also indicated that signal processing on data could be achieved without any knowledge of the data model, by letting the intrinsic data geometry emerge through a natural process of affinity diffusion. Modern sensor systems such as radar, hyperspectral, MRI and others actually do not measure images but much more elaborate vectors, the images are built to allow understanding and further processing, in reality we should let the intrinsic geometry of the measurements participate in the information extraction. Such an approach has been developed by our team for hyperspectral imaging.

We also observe that in the context of compressed sensing where the sensor inputs are randomly encoded. The projection into a random coded subspace while maintaining the relative affinity of the original data points permits rebuilding the data geometry by tools described above.

7. ACKNOWLEDGEMENTS

RC is partially supported by DARPA and NSF. MM is partially supported by NSF Grant DMS 0512050. SWZ is partially supported by AFRL.

REFERENCES

1. S. Lafon, *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, Dept of Mathematics & Applied Mathematics, 2004.
2. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comp. Harm. Anal.* , 2006. To appear.

3. R. Coifman and S. Lafon, "Geometric harmonics: a novel tool for multiscale out-of-sample extension of empirical functions," *Appl. Comp. Harm. Anal.* , 2006. To appear.
4. R. R. Coifman and M. Maggioni, "Diffusion wavelets," *Tech. Rep. YALE/DCS/TR-1303, Yale Univ., Appl. Comp. Harm. Anal.* , Sep. 2004. To appear.
5. R. R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data. part i: Diffusion maps," *Proc. of Nat. Acad. Sci.* , pp. 7426–7431, May 2005.
6. R. R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, and S. Zucker, "Geometric diffusions as a tool for harmonic analysis and structure definition of data. part ii: Multiscale methods," *Proc. of Nat. Acad. Sci.* , pp. 7432–7438, May 2005.
7. R. R. Coifman and M. Maggioni, "Multiscale data analysis with diffusion wavelets," Tech. Rep. YALE/DCS/TR-1335, Dept. Comp. Sci., Yale University, September 2005.
8. A. D. Szlam, *Non-stationary analysis of datasets and applications.* PhD thesis, Yale University, Dept of Mathematics & Applied Mathematics, 2006.
9. R. R. Coifman, M. Maggioni, S. W. Zucker, and I. G. Kevrekidis, "Geometric diffusions for the analysis of data from sensor networks," *Curr Opin Neurobiol* **15**, pp. 576–84, October 2005.
10. M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation* **6**, pp. 1373–1396, June 2003.
11. E. Causevic, R. R. Coifman, R. Isenhart, A. Jacquin, E. R. John, M. Maggioni, L. S. Prichep, and F. J. Warner, "QEEG-based classification with wavelet packets and microstate features for triage applications in the ER," Oct 2005. ICASSP 05.
12. M. Maggioni, J. C. Bremer Jr, R. R. Coifman, and A. D. Szlam, "Biorthogonal diffusion wavelets for multiscale representations on manifolds and graphs," August 2005. Proc. SPIE Wavelet XI.
13. M. Maggioni, A. D. Szlam, , R. R. Coifman, and J. C. Bremer Jr, "Diffusion-driven multiscale analysis on manifolds and graphs: top-down and bottom-up constructions," August 2005. Proc. SPIE Wavelet XI.
14. S. Mahadevan and M. Maggioni, "Value function approximation with diffusion wavelets and laplacian eigenfunctions," in *University of Massachusetts, Department of Computer Science Technical Report TR-2005-38; Proc. NIPS 2005*, 2005.
15. M. Maggioni and S. Mahadevan, "Fast direct policy evaluation using multiscale analysis of markov diffusion processes," in *University of Massachusetts, Department of Computer Science Technical Report TR-2005-39; submitted*, 2005.

Surface Geometric Constraints for Stereo in Belief Propagation

Gang Li and Steven W. Zucker

Department of Computer Science
Yale University
{gang.li, steven.zucker}@yale.edu

Abstract

Belief propagation has been shown to be a powerful inference mechanism for stereo correspondence. However the classical formulation of belief propagation implicitly imposes the frontal parallel plane assumption in the compatibility matrix for exploiting contextual information, since the priors perfer no depth (disparity) change in surrounding neighborhoods. This results in systematic errors for slanted or curved surfaces. To eliminate these errors we propose to use contextual information geometrically, and show how to encode surface differential geometric properties in the compatibility matrix for stereo correspondence. This enforces consistency for both depth and surface normal, extending the traditional formulation beyond consistency for (constant) depth. With such geometric contextual information, the belief propagation algorithm shows dramatic improvement on generic non-frontal parallel scenes. Several such examples are provided.

1. Introduction

In recent years both belief propagation[28, 29] and graph cuts[5, 17, 18] have been proposed to solve the stereo correspondence problem. They have achieved great success as their variants keep topping the comparison chart for the Middlebury dataset [29, 27]. However a closer look at these image pairs indicates that they are quite limited in terms of the surface types represented. The ground truth disparity shows that every object has very limited (or no) disparity change, indicating every single object can be well described by a (combination of) frontal parallel plane(s). Most importantly this limitation has been exploited algorithmically. When using contextual information, both belief propagation[28, 29] and graph cuts[5, 17] use either the Potts energy model or the truncated linear (quadratic) energy model, which implicitly use the frontal parallel plane assumption: namely that, within a region under consideration, position disparity (or depth) is constant with respect to the rectified stereo image pair.

However the real world consists of objects of much richer geometry than frontal parallel planes. Can these algorithms handle slanted or curved surfaces? We show that current formulations cannot handle such scenes well. Our goal in this paper is to move beyond those limitations for the belief propagation algorithm. Others have attempted more general surface types. Slanted planar surfaces are explicitly modeled for segmented regions in [3], where segmentation and correspondence are iteratively obtained from the multiway-cut algorithm [5]; this has been generalized to curved surfaces [23]. In [24] a slanted scanline algorithm is developed for slanted planar surfaces. To our knowledge there are no other attempts at using surface differential geometry directly for general, smooth surfaces.

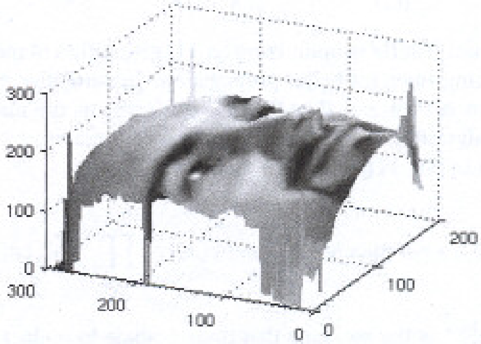


Figure 1. 3D reconstruction of a face from a pair of stereo images. Our algorithm can achieve accurate geometric modeling of such smooth curved surfaces. Since surface normals are changing smoothly almost everywhere, it follows that the tangent planes are mostly not frontal parallel planes. Therefore it is necessary to develop richer geometry in computing the compatibilities in belief propagation than what has been currently used.

Using belief propagation as our algorithmic framework, we argue that for general surface types differential geo-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE

metric properties have to be taken into consideration when using contextual information. In particular, we derive the compatibilities using both position disparity (or depth) and disparity derivatives (surface normal) for general planar and smooth curved surfaces and illustrate their use on scenes with slanted surfaces and faces. Comparison results with the traditional formulation of compatibilities [29, 28] clearly demonstrate the importance of using both positional (zeroth-order) disparity (or depth) and surface normal (first-order disparities). Fig. 1 is a preview of our face reconstruction example (Fig. 8). Such accurate surface reconstructions are necessary for applications in 3D modeling, computer graphics, facial expression recognition, and surgical planning.

2. Problem Formulation

With a rectified stereo pair [9, 13], stereo correspondence can be formulated as the estimation of a random variable x_k for every node k in a Markov Random Field (MRF), with x_k the (positional) disparity at k. (Notation follows [11, 29]). Further assuming a pair-wise MRF, let $\Psi(x_i, x_j)$ denote the compatibility function, which encodes the compatibility between two immediate neighboring nodes i and j, and $\Phi(x_k, y_k)$ (also shortened as $\Phi(x_k)$) denote local evidence that variable x_k is consistent with observation y_k. Then the joint probability of this MRF is:

$$
\begin{aligned}
&P(x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N) \\
&= \prod_{(i,j)} \Psi_{ij}(x_i, x_j) \prod_k \Phi(x_k, y_k)
\end{aligned}
\tag{1}
$$

Different criteria in optimizing eq. (1) give different message passing rules for belief propagation. In particular, the Maximum A Posterior (MAP) estimator gives us the max-product algorithm. Specifically, messages at iteration $t+1$ are given by[28, 11]:

$$
m_{ij}^{t+1}(x_j) \leftarrow \alpha \max_{x_i} \Psi_{ij}(x_i, x_j)\Phi(x_i) \prod_{k \in N(i) \setminus j} m_{ki}^t(x_i) \tag{2}
$$

where m_{ij}^{t+1} is the message that node i sends to node j at iteration $t+1$, $N(i) \setminus j$ is the set of nodes neighboring node i except node j itself, and α is a normalization term.

The belief b_i at node i is then computed as:

$$
b_i(x_i) \leftarrow \alpha\Phi(x_i) \prod_{k \in N(i)} m_{ki}(x_i) \tag{3}
$$

The MAP solution at node i is:

$$
x_i^{MAP} = \arg \max_{x_k \in \{d_1, \ldots, d_L\}} b_i(x_k) \tag{4}
$$

Alternatively one can obtain the Minimum Mean Squared Error (MMSE) estimate [11], which requires to change $\max_{x_i}$ in eq. (2) to $\sum_{x_i}$, (thus called the sum-product algorithm), and accordingly compute the solution as $x_i^{MMSE} = \sum_{x_k=d_1}^{d_L} x_k b_i(x_k)$.

For direct comparison with other belief propagation stereo algorithms[29, 28], we use the max-product message updating rule to find the MAP estimate. It has been shown[29] that this is the same problem that Graph Cuts algorithms have been solving [5]. Our focus in this paper is to develop the compatibilities geometrically, however efficient techniques[10] could be used to speed up our implementation.

2.1. Traditional Compatibilities Ψ_{ij}

The most common models for $\Psi_{ij}(x_i, x_j)$ both prefer no disparity change between two neighboring nodes.

2.1.1 Potts Energy Model Derived Ψ_{ij}

First, the compatibilities in [29] are defined as:

$$
\Psi_{ij}^{Potts}(x_i, x_j) = exp(-\frac{V(x_i, x_j)}{\sigma_V}) \tag{5}
$$

which is derived from the Potts energy model with

$$
V(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ \rho_I(|I_i - I_j|) & \text{otherwise} \end{cases}
$$

where

$$
\rho_I(|I_i - I_j|) = \begin{cases} C_1 \times \lambda_1 & \text{if } |I_i - I_j| < T \\ \lambda_1 & \text{otherwise} \end{cases}
$$

with T a threshold, λ_1 a penalty term for having different disparities, and C_1 a penalty term that increases the penalty when the image intensity difference is small. T, λ_1, and C_1 are constant parameters. The Potts model assumes that labelings should be piecewise constant.

2.1.2 Truncated Linear Energy Model Derived Ψ_{ij}

Second, the compatibilities can be derived from the truncated linear energy model, where the cost increases linearly as a function of the difference between two labels, up to a threshold and then stays as a constant: $V(x_i, x_j) = \min(\lambda_2|x_i - x_j|, C_2)$, with λ_2 and C_2 constant parameters.

For stereo,[28] has derived the compatibilities from such a truncated linear potential function using the Total Variance (TV) model:

$$
\Psi_{ij}^{TL}(x_i, x_j) = (1 - \epsilon_p)exp(-\frac{|x_i - x_j|}{\sigma_p}) + \epsilon_p \tag{6}
$$

with ϵ_p a small constant and σ_p a constant. ϵ_p and σ_p together control the shape of the robust potential function.

Clearly both of these compatibility matrices prefer no disparity change between two neighboring nodes, i.e. they implicitly use the frontal parallel plane assumption.

3. Differential Geometric Constraints in Belief Propagation

We now extend the compatiblities to include surface differential geometric constraints. As pointed out by [4, 22], using finite differences one has to consider cliques of three neighboring nodes in the compatibility matrix for the plate model, which accounts for an underlying slanted planar surface model. But such an endeavor quickly makes the problem computationally infeasible.

To use differential properties, at first consideration the computational complexity issues would seem to multiply. Not only are many disparity (depth) labels required, but derivatives must be labels as well. In this section we show how to attach differential properties to what we describe as "floating" disparities (labels), such that the problem formulation remains the same (i.e. pair-wise MRF), but great improvements can be achieved by considering higher-order differential geometric constraints for surface smoothness.

3.1. "Floating" Disparities

We now describe a different interpretation of the classical formulation which allows us to make a small but necessary change to the problem formulation. Fig. 2 shows that node i is connected to node j and sends message m_{ij} to node j. Random variables x_i (at node i) and x_j (at node j) take values from L discretized disparities $\{ d_1, d_2, \ldots, d_L \}$. The connection (edge) between two labels (d_i and d_j) contributes to the whole message m_{ij} from node i to node j.

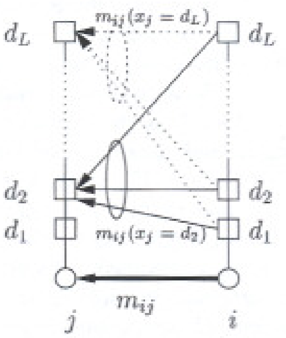


Figure 2. 1D illustration of message passing. Nodes i and j have L possible labels $\{ d_1, \ldots, d_L \}$. Message m_{ij} from node i to node j is a vector encoding the "support" each label at j receives from all possible labels at i.

For detailed geometric modeling tasks this is usually not enough. Directly increasing the number of discretized disparities (using subpixel disparity levels) quickly makes it computationally infeasible, especially for scenes with large disparity range. As an alternative approach, we still use L disparities, which are initialized to be integer disparities but can then be changed to continuous (floating) disparity based on interpolation, similarly to deformable models [15] in spirit. In other words, the locations of the labels can be adapted according to initial measurements so that the initial lattice of disparity labels is adapted to the scene structure. In particular, we compute a normalized SSD score using a deformed window approach as in [7]. A direction set method[26] is used to find the floating point $\{ d, \frac{\partial d}{\partial u}, \frac{\partial d}{\partial v} \}$ that gives the best normalized SSD score. Suppose $d \in [d_k, d_{k+1})$; then we change the disparity label structure to let d_k "float" to d. In practice for each node i we can perform such computations at several local minima obtained from initial integer SSD. The result is a deformable disparity structure which encodes local measurements based on a deformed window SSD. Note that the differential properties (first order disparity derivatives) are also encoded at each state d_i for node i. Also note that time complexity in computing messages (eq.(2)) remains the same.

3.2. Differential Geometric Constraints in Compatibility Matrix Ψ_{ij}

To illustrate how to encode surface geometry in the computation of messages we walk through the specific computations and point out where geometry comes in. m_{ij} (eq.(2)) is the message from node i to node j. It is a vector which contains the "support" that individual labels in node j receive from all labels in node i. For each state in node j it is computed as:

$$\begin{aligned} & m_{ij}^{t+1}(x_j = d_l) \\ & \quad \leftarrow \alpha \max_{x_i} \Psi_{ij}(x_i, d_l)\Phi(x_i) \prod_{k \in N(i) \setminus j} m_{ki}^t(x_i) \end{aligned} \tag{7}$$

Fig.2 illustrates $m_{ij}(x_j = d_2)$ and $m_{ij}(x_j = d_L)$.

3.2.1 Slanted Planar Surfaces

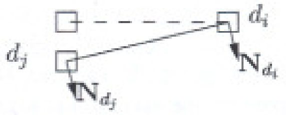


Figure 3. 1D illustration: A general planar surface model at d_i imposes constraints that d_j lies on the same plane (in solid lines) and has the same surface normal. Also shown is the frontal parallel plane at d_i (in dotted lines).

For node i with label d_i, a general planar surface model at d_i is $d(u, v) = d_i + \frac{\partial d_i}{\partial u}(u - u_i) + \frac{\partial d_i}{\partial v}(v - v_i)$, representing the fact that contextual information prefers a neighboring label d_j to lie on the same planar surface (possibly non-frontal parallel) and should have the same surface normal. Fig. 3 shows this point. The compatibility using such a coplanar model is:

$$\Psi_{ij}(x_i = d_i, x_j = d_j) = \tag{8}$$
$$((1-\epsilon_p)exp(-\frac{|d_j - d_i - \frac{\partial d_i}{\partial u}(u_j - u_i) - \frac{\partial d_i}{\partial v}(v_j - v_i)|}{\sigma_p})$$
$$+\epsilon_p)((1-\epsilon_N)exp(-\frac{\|\mathbf{N}_{d_j} - \mathbf{N}_{d_i}\|^2}{\sigma_N}) + \epsilon_N)$$

where $\mathbf{N}$ is the surface normal in the disparity space and can be computed as $\mathbf{N} = \frac{(-\frac{\partial d}{\partial u}, -\frac{\partial d}{\partial v}, 1)^T}{\sqrt{(\frac{\partial d}{\partial u})^2 + (\frac{\partial d}{\partial v})^2 + 1}}$. If the cameras are further assumed to be calibrated, surface normals in Euclidean space $\mathbf{N} = \frac{(-z_x, -z_y, 1)}{\sqrt{1 + z_x^2 + z_y^2}}$ can also be used. It is derived from $z(u,v) = \frac{\alpha b}{d(u,v)}$, which leads to $z_x = -\frac{\alpha b}{d^2}\frac{\partial d}{\partial u}\frac{\alpha}{f}$ and $z_y = -\frac{\alpha b}{d^2}\frac{\partial d}{\partial v}\frac{\alpha}{f}$, with b the stereo baseline, α focal length in pixels and f focal length in physical unit. Note that this compatibility measure contains the frontal parallel plane model as a special case. When the underlying model at d_i is frontal parallel plane, $\frac{\partial d_i}{\partial u} = \frac{\partial d_i}{\partial v} = 0$, and $\mathbf{N}_{d_i} = \mathbf{N}_{d_j}$, the above compatibility simplifies to eq.(6).

3.2.2 Curved Surfaces

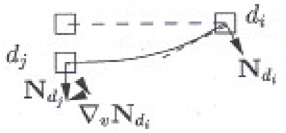


Figure 4. 1D illustration: A smooth curved surface model at d_i imposes constraints that d_j lies on the same surface and has surface normal $\mathbf{N}_{d_i} + \nabla_v \mathbf{N}_{d_i}$. Also shown is the frontal parallel plane and the tangent plane at d_i (in dotted lines).

For curved surfaces not only the depth (positional disparity) changes in the neighborhood, but the surface normal also changes. To explicitly take this into account in the compatibilities we have to consider the covariant derivative of the surface normal, $\nabla_v \mathbf{N}_{d_i}$, which encodes the change of surface normal for a (displacement) tangent vector $\mathbf{v}$ at d_i. Fig. 4 illustrates this point. $\nabla_v \mathbf{N}_{d_i}$ can be computed using the shape operator (or second fundamental form) [8, 16, 25, 6], for a tangent vector $\mathbf{v}$ in the tangent plane $T_p(M)$. The computation will involve second order disparity derivatives, which can be either estimated initially together with $\{d, \frac{\partial d}{\partial u}, \frac{\partial d}{\partial v}\}$, as attempted in [7], or more preferably by a local fitting procedure over the initial estimates in Euclidean space. Space limitation prevents us from a detailed discussion, but our analysis indicates that the latter approach is the numerically stable one for dealing with higher order differential properties. We use this one in our computations. For detailed discussions on the geometric computation, see [21].

The compatibility using a quadratic approximation as the local surface model of a curved surface is:

$$\Psi_{ij}(x_i = d_i, x_j = d_j) = \tag{9}$$
$$((1-\epsilon_p)exp(-\frac{|d_j - d_i - \frac{\partial d_i}{\partial u}(u_j - u_i) - \frac{\partial d_i}{\partial v}(v_j - v_i)|}{\sigma_p})$$
$$+\epsilon_p)((1-\epsilon_N)exp(-\frac{\|\mathbf{N}_{d_j} - \mathbf{N}_{d_i} - \nabla_v \mathbf{N}_{d_i}\|^2}{\sigma_N}) + \epsilon_N)$$

Note that when the underlying surface is a planar surface $\nabla_v \mathbf{N}_{d_i} \equiv 0$, the above compatibility simplifies to eq. (8).

4. Experimental Results

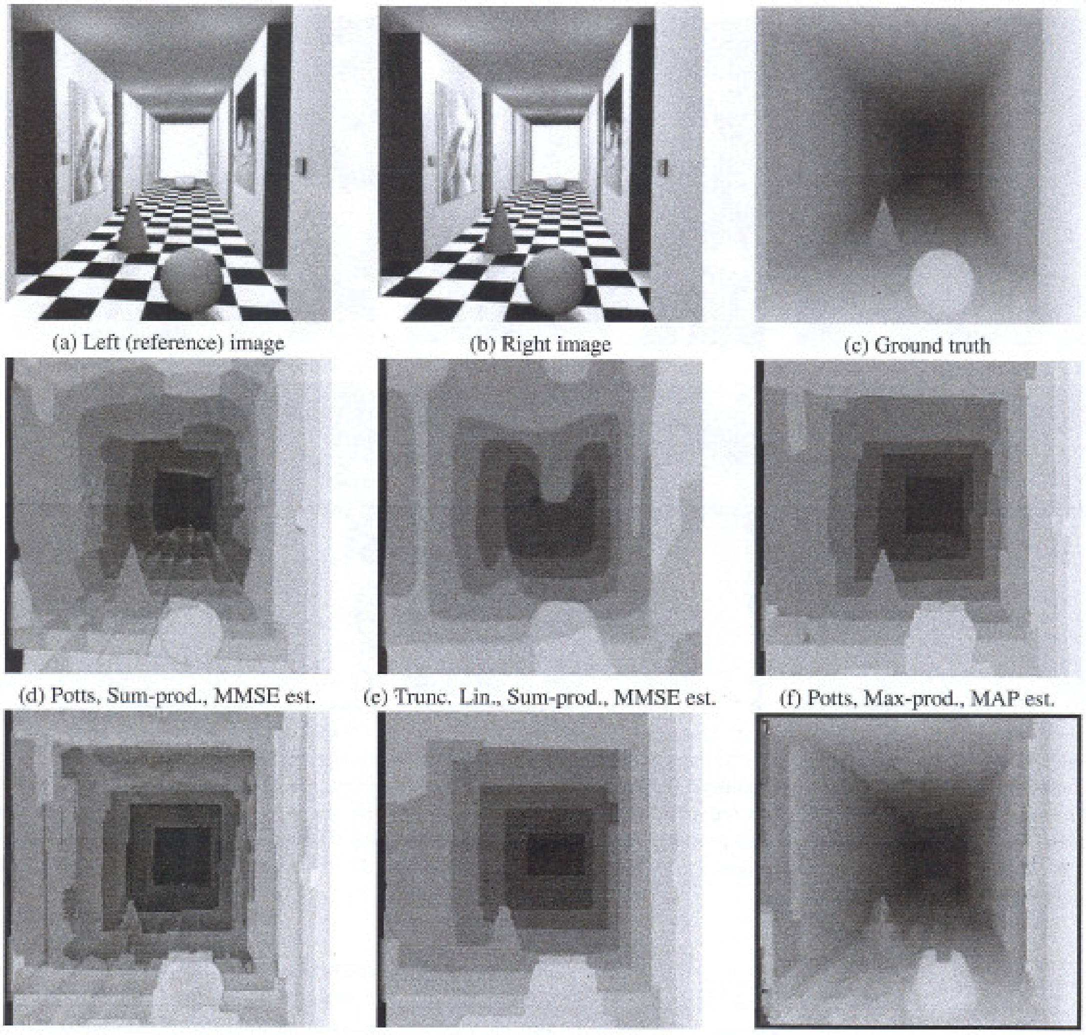
In this section we describe representative experimental results on various scenes, with some quantitative error analysis. In particular we compare results using our formulation of compatibities with using compatibilities that have been used in the literature [29, 28], which demonstrates that our new formulation indeed improves the performance of belief propagation algorithm on generic non-frontal parallel scenes.

Local evidence $\Phi(x_i)$ is computed using similar formula as in [29]:

$$\Phi(x_i) = exp(-\frac{D_i(x_i)}{\sigma_d})$$

with $D_i(x_i)$ the truncated data cost $D_i(x_i) = \min(|I_l(u,v) - I_r(u - x_i, v)|, C)$ at node i (i.e. pixel (u,v)) with disparity x_i, as in [29, 10]. For results with traditional compatibilities (eq. (5) and eq. (8)), the Birchfield-Tomasi technique [2] is used in computing the image intensity difference in the above data cost, the same as in [29, 28]. We use $C = 50$ and $\sigma_d = 50$ in the experiments.

The first example (Fig. 5) is the synthetic "Corridor" pair [12] with ground truth (courtesy of University of Bonn). The image size is 256x256 pixels with a disparity range of 11 pixels. The underlying scene structure consists of slanted planar surfaces. Fig.5(c) shows the ground truth disparity map. Fig. 5(d-i) shows the results using different compatibilities. In particular, Fig.5(d) is the result using Potts energy model derived compatibilities (eq. (5)), with sum-product message updating and the MMSE solution; Fig.5(e) is with Truncated Linear energy model derived compatibilities (eq. (6)) with sum-product message updating and the MMSE solution; Fig.5(f) is with Potts energy model derived compatibilities (eq. (5)), with max-product message updating and the MAP solution (the same as in


(a) Left (reference) image (b) Right image (c) Ground truth

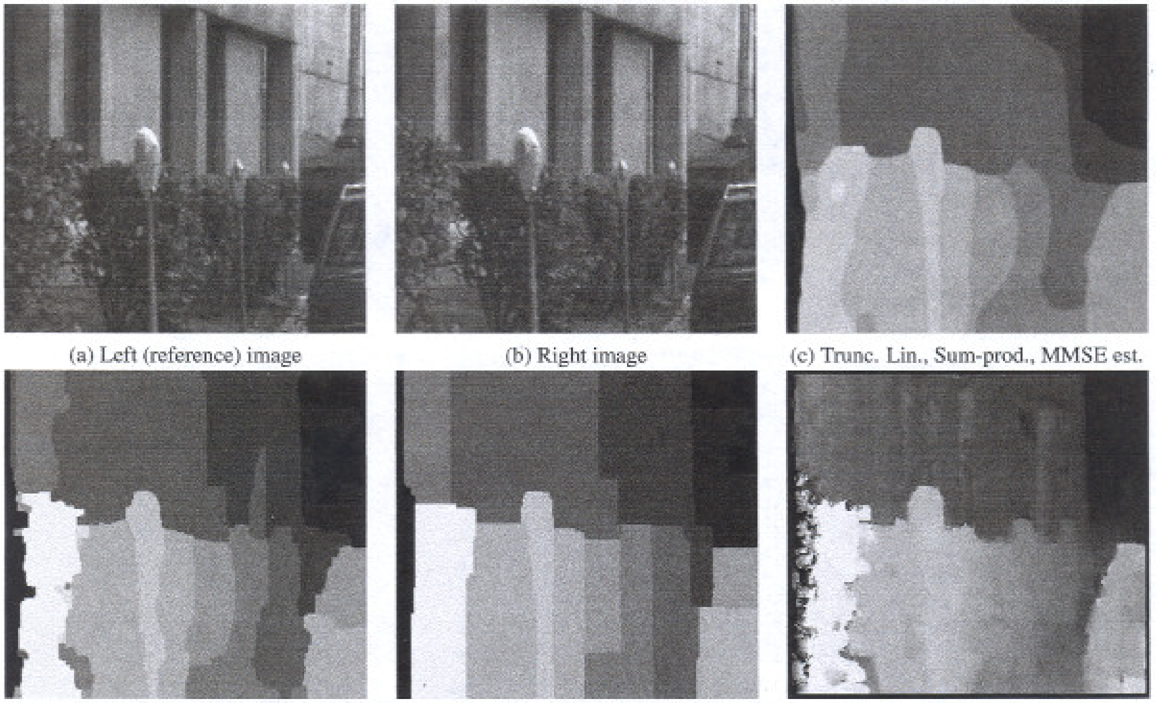
(d) Potts, Sum-prod., MMSE est. (e) Trunc. Lin., Sum-prod., MMSE est. (f) Potts, Max-prod., MAP est.

(g) Potts, Max-prod., MAP+subpixel (h) Trunc. Lin., Max-prod., MAP est. (i) Our compat., Max-prod., MAP est.

Figure 5. (a)(b) Left (reference) and right images. (c) Ground truth disparity map. (d) Potts model derived compatibilities (eq. (5)), Sum-product, MMSE estimate. (e) Truncated linear energy model derived compatibilities (eq. (6)), Sum-product, MMSE estimate. (f) Potts model derived compatibilities (eq. (5)), Max-product, MAP estimate. (g) Potts model derived compatibilities (eq. (5)), Max-product, MAP estimate+subpixel refinement. (h) Truncated linear energy model derived compatibilities (eq. (6)), Max-product, MAP estimate. (f) Our general planar surface model derived compatibilities (eq. (8)) proposed in this paper, Max-product, MAP estimate. Using other compatibilities one obtains stepwise scalloped patterns because of the frontal parallel plane assumption. On the other hand our result has gradual smooth disparity change, indicating the correct reconstruction. Error statistics in Fig. 6.

[29]), as well as a subpixel interpolated version based on the SSD cost of a 11x11 window in Fig.5(g); Fig.5(h) shows the result using Truncated Linear energy derived compatibilites (eq. (6)), with max-product message updating and the MAP solution (the same as in [28]); and finally in Fig.5(i) we show our result using the general planar surface model derived compatibilities (eq. (8)) proposed in this paper, with max-product message updating and the MAP solution. We choose a fixed set of parameters throughout the experimental section. In particular, $\sigma_V = 50$, $C_1 = 2$, $\lambda_1 = 50$, $T = 4$, and $\epsilon_p = 0.05$, $\sigma_p = 0.6$, $\epsilon_N = 0.05$, $\sigma_N = 0.4$.

In Fig.6 we show the error statistics using the taxonomy package [27]. In particular we compute the percentage of bad matching pixels with absolute disparity error larger than different thresholds ranging from 0.25–1.50 pixels. In our result, we achieve better error statistics because we explicitly model 3D surface geometry. The computational complexity of message updating with the new compatibilities



(a) Left (reference) image (b) Right image (c) Trunc. Lin., Sum-prod., MMSE est.

(d) Potts, Max-prod., MAP est. (e) Trunc. Lin., Max-prod., MAP est. (f) Our compat., Max-prod., MAP est.

Figure 7. (a)(b) Left (reference) and right images. (c) Disparity map using Truncated Linear energy model derived compatibilities (eq. (6)), sum-product message updating and the MMSE solution. (d) Potts model derived compatibilities (eq. (5)), Max-product, MAP estimate. (e) Truncated Linear energy model compatibilities (eq. (6)), Max-product, MAP estimate. (f) Our general planar surface model derived compatibilities (eq. (8)) proposed in this paper, Max-product, MAP estimate. Once again using other compatibilities one obtains stepwise scalloped patterns because of the frontal parallel plane assumption being used. On the other hand our result has gradual smooth disparity change, indicating the correct reconstruction.

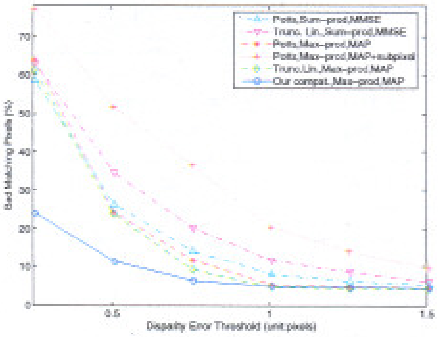
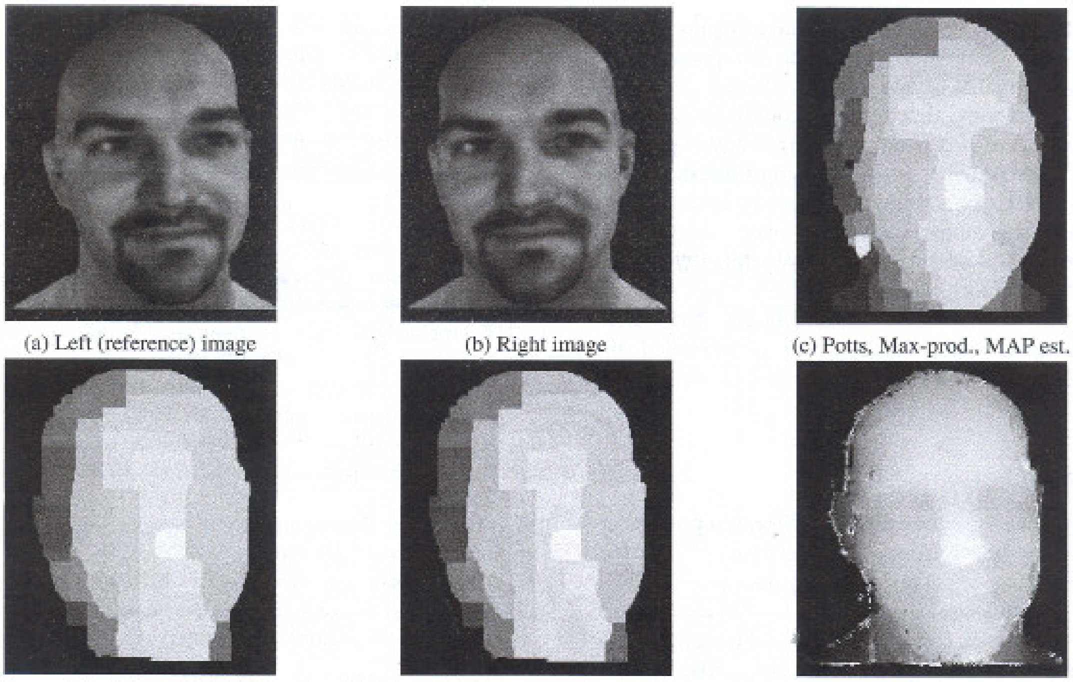


Figure 6. Error statistics for Corridor pair: shown is the percentage of bad matching pixels using the taxonomy package [27] at 5 different thresholds ranging from 0.25–1.50 pixels. Our result has better error statistics for such slanted planar surfaces because our compatibilities explicitly encode such surface geometry.

(eq. (8)) is the same as with the standard compatibilities (eq (6)). But in practice the standard compatibilities are pre-computed and stored in a lookup table, while the new compatibilities are explicitly computed (although they can similarly be stored in a lookup table). Thus our algorithm's running time increases over the standard max-product belief propagation algorithm with compatibilities as in eq. (6). For this image pair we observe an increase from 10 seconds to 16 seconds for each iteration of message updating on a 2.4GHz CPU. The proposed compatibilities will also require computational overheads to obtain the initial differential properties; we use a 11x11 deformed window (as in [7, 19]) and have observed a running time of about 8 milliseconds per pixel (node).

The second example is the "Parking meter" pair from the well-known JISCT database. The image size is 256x240 pixels with a disparity range of 10 pixels. Fig. 7 shows the results using different compatibilities. Specifically, Fig.7(c) shows the result using the Truncated Linear energy model derived compatibilities (eq. (6)), with sum-product message updating and the MMSE solution; Fig. 7(d) is the result using the Potts energy model derived compatibilities (eq. (5)), with max-product message updating and the MAP soultion



(a) Left (reference) image (b) Right image (c) Potts, Max-prod., MAP est.

(d) Trunc. Lin., Max-prod., MAP est. (e) Trunc. Lin., Max-prod., MAP+subpixel (f) Our compat., Max-prod., MAP est.

Figure 8. (a)(b) Left (reference) and right images. (c) Potts model derived compatibilities (eq. (5)), Max-product, MAP estimate. (d) Truncated linear energy model derived compatibilities (eq. (6)), Max-product, MAP estimate. (d) Truncated linear energy model derived compatibilities (eq. (6)), Max-product, MAP estimate+subpixel refinement. (f) Our smooth curved surface model derived compatibilities (eq. (9)) proposed in this paper, Max-product, MAP estimate. As in the previous examples using other compatibilities one obtains stepwise scalloped patterns because of the frontal parallel plane assumption. On the other hand our result has gradual smooth disparity change, indicating the correct reconstruction.

(the same as in [29]); and Fig.7(e) is from Truncated Linear energy model dervied compatibilites (eq. (6)), with max-product message updating and the MAP solution (the same as in [28]); finally Fig.7(f) is our result using the general planar surface model derived compatibilities (eq. (8)) proposed in this paper, with max-product message passing and the MAP solution. Running time with the proposed compatibilities is 11 seconds per iteration of message updating.

The third example is a stereo pair of a human face (Fig.8). Ground truth data (3D geometry and texture map) were obtained from the $Cyberware^{TM}$ laser scanner dataset. The true disparity map is then computed. The stereo pair has a baseline of 6cm and focal length 1143 pixels. The human head ranges from 26.5cm to 53.5cm in front of the camera. The original image size is 1024x768 pixels but then subsampled to 256x192 pixels and further cropped to 160x192 pixels with a disparity range of 31 pixels.

Fig.8 shows the results using different compatibilities. In particular, Fig.8(c) is from the Potts energy model derived compatibilities (eq. (5)), with max-product message updating and the MAP solution (the same as in [29]); Fig.8(d) is from Truncated Linear energy model derived compatibilites (eq. (6)), with max-product message updating and the MAP solution (the same as in [28]), as well as a subpixel interpolated version based on the SSD cost of a 11x11 window in

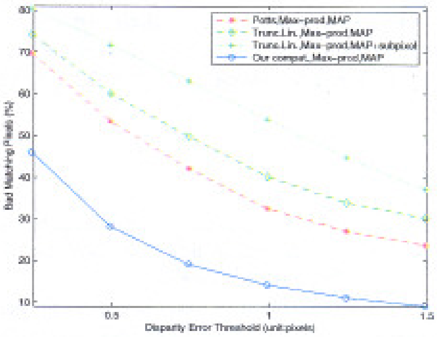


Figure 9. Error statistics for Face pair: shown is the percentage of bad matching pixels using the taxonomy package [27] at 5 different thresholds ranging from 0.25–1.50 pixels. Our result has better error statistics for such smooth curved surfaces because our compatibilities explicitly encode such surface geometry.

Fig.8(e); and finally in Fig.8(f) is our result using the curved surface model derived compatibilities (eq. (9)) proposed in this paper, also with max-product message updating and the MAP solution. Running time of our algorithm is about 80 seconds per iteration of message updating. For this pair the occlusion is detected using a combination of thresholds on the deformed SSD score, belief, and local evidence. Similarly as in the first example, Fig.9 is the error statistics. We achieve better error statistics because we explicitly model such 3D curved surface geometry.

Note that the problem as formulated in Fig. 2 can also be solved using relaxation labeling [14], as employed for feature-based stereo in [20] and for texture flow analysis in [1].

5. Conclusion

In this paper we introduce surface differential geometric constraints to the belief propagation algorithm. In particular, we derive the compatibilities using both position disparity (or depth) and surface normal for slanted and curved surfaces, and illustrate their use. Such compatibilites extend traditional belief propagation to handle generic surface types. The result is an improved belief propagation algorithm that can perform well on slanted or curved surfaces. Experimental results demonstrate the importance of incorparating surface differential geometry with powerful inference algorithm.

Acknowledgments

We thank Richard Szeliski, Daniel Scharstein, and Marshall Tappen for helpful explanations of their software packages [27, 29], as well as the anonymous reviewers for detailed comments. Research supported by AFRL, AFOSR, and DARPA.

References

[1] O. Ben-Shahar and S. W. Zucker. The perceptual organization of texture flow: A contextual inference approach. *IEEE Trans. on PAMI*, 25(4):401–417, 2003.

[2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. on PAMI*, 20(4):401–406, 1998.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proc. ICCV*, 1999.

[4] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, 1987.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on PAMI*, 23(11):1222–1239, 2001.

[6] R. Cipolla and P. Giblin. *Visual Motion of Curves and Surfaces*. Cambridge Univ. Press, 2000.

[7] F. Devernay and O. D. Faugeras. Computing differential properties of 3-d shapes from stereoscopic images without 3-d models. In *Proc. CVPR*, 1994.

[8] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.

[9] O. Faugeras. *Three-Dimensional Computer Vision*. The MIT Press, 1993.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. In *Proc. CVPR*, 2004.

[11] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *IJCV*, 40(1):25–47, 2000.

[12] T. Frohlinghaus and J. M. Buhmann. Regularizing phase-based stereo. In *Proc. of ICPR*, 1996.

[13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, 2000.

[14] R. A. Hummel and S. W. Zucker. On the foundations of relaxation labeling processes. *IEEE Trans. on PAMI*, 5(3):267–287, 1983.

[15] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, pages 321–331, 1988.

[16] J. J. Koenderink. *Solid Shape*. The MIT Press, 1990.

[17] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proc. ICCV*, 2001.

[18] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *Proc. ECCV*, 2002.

[19] G. Li and S. W. Zucker. Stereo for slanted surfaces: First order disparities and normal consistency. In *Proc. EMMCVPR, LNCS 3757*, 2005.

[20] G. Li and S. W. Zucker. Contextual inference in contour-based stereo correspondence. *IJCV, in press*, 2006.

[21] G. Li and S. W. Zucker. Differential geometric consistency extends stereo to curved surfaces. In *Proc. ECCV*, 2006.

[22] S. Z. Li. *Markov Random Field Modeling in Image Analysis*. Springer-Verlag, 2001.

[23] M. H. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. *IEEE Trans. on PAMI*, 26(8):1073–1078, 2004.

[24] A. S. Ogale and Y. Aloimonos. Stereo correspondence with slanted surfaces: Critical implications of horizontal slant. In *Proc. CVPR*, 2004.

[25] B. O'Neill. *Elementary Differential Geometry*. Academic Press, second edition, 1997.

[26] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipies in C*. Cambridge University Press, second edition, 1992.

[27] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002.

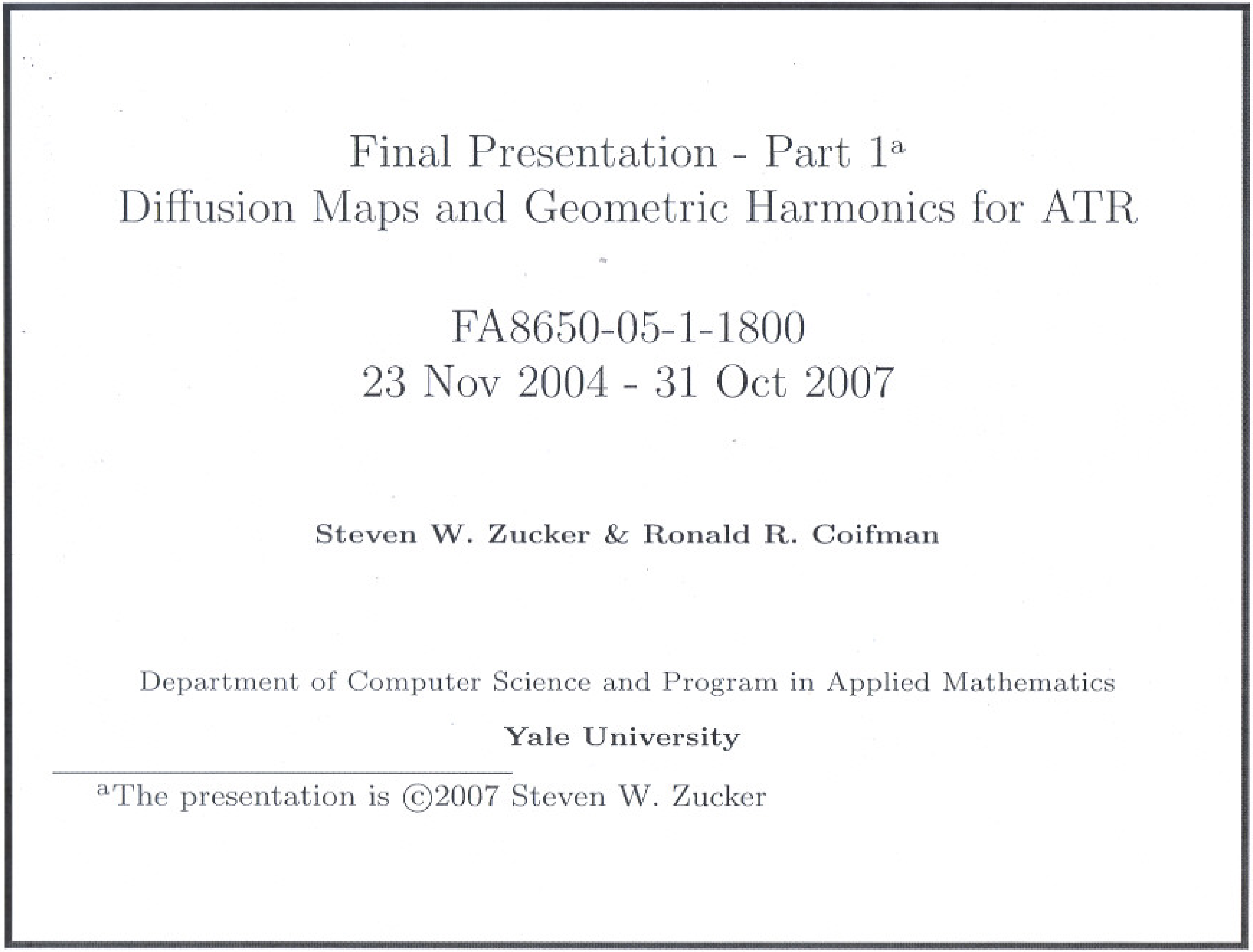
[28] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. on PAMI*, 25(7):787–800, 2003.

[29] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proc. ICCV*, 2003.

0-7695-2597-0/06 $20.00 © 2006 IEEE

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government."

"This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."



Final Presentation - Part 1[a]
Diffusion Maps and Geometric Harmonics for ATR

FA8650-05-1-1800

23 Nov 2004 - 31 Oct 2007

Steven W. Zucker & Ronald R. Coifman

Department of Computer Science and Program in Applied Mathematics

Yale University

[a] The presentation is ©2007 Steven W. Zucker

Supported by : AFRL

Thanks to :

Andreas Glaser.
Patrick Huggins.
Yosi Keller.
Gang Li.
Edo Liberty.
Mauro Maggioni

Goals:

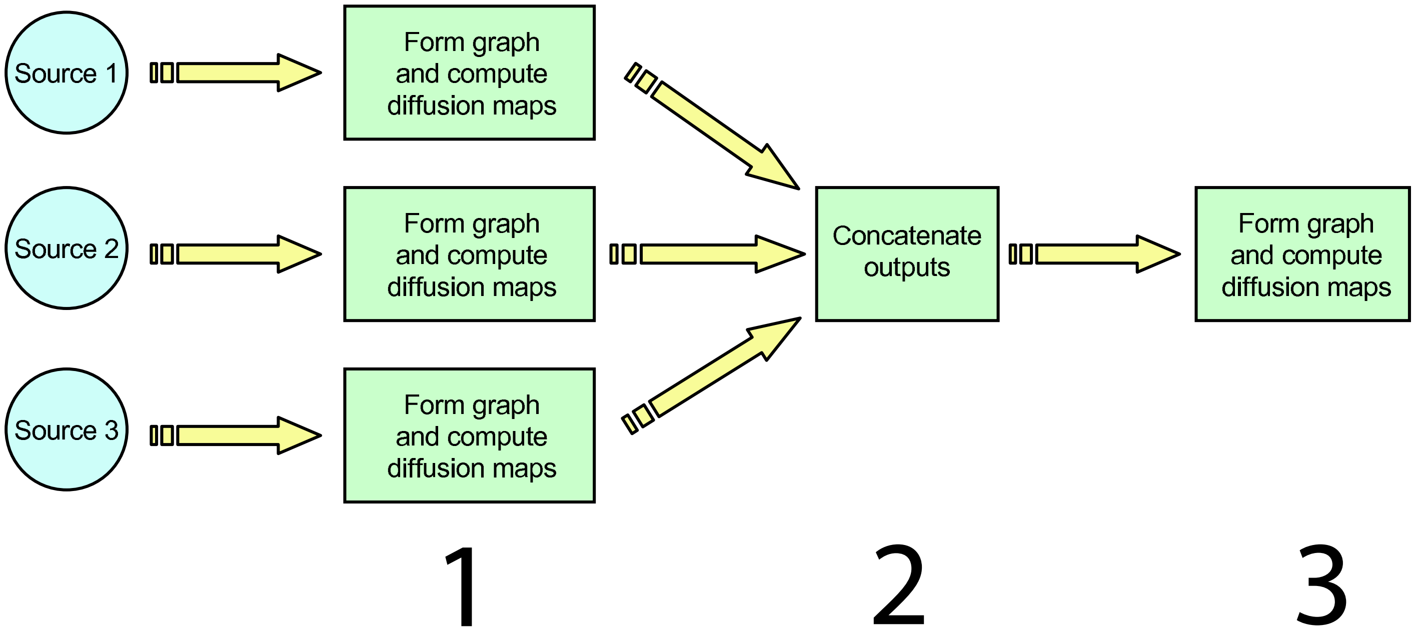
- ATR Pattern Classification from Multiple Data Sources
- Representation of Information
- Dimensionality-reduction Framework
  - Based on Diffusion Maps (”non-linear pca”)
  - Purely Data Driven
  - Reveal Intrinsic Data Geometry
- Fusion of Data Sources
- Simple → Complex Features
- “Symbolic” Features

Accomplishments:

- Dimensionality-reduction Framework for Data Fusion
- Fusion of boundary/texture/color data for improved segmentation
- Fusion of Auditory and Visual Data for improved classification
- (Fusion of left/right stereo pairs): advanced geometry
- (Embeddings of Symbolic Data): MMPI

Overview of Talk:

- Dimensionality-reduction Framework
- Results: Fusion of Auditory and Visual Data for improved classification
- Example: Fusion of boundary/texture/color data for improved segmentation
- Overview of stereo system.
- Overview of color projections.
- Overview of MMPI-2 results.



Source 1

Source 2

Source 3

Form graph and compute diffusion maps

Form graph and compute diffusion maps

Form graph and compute diffusion maps

Concatenate outputs

Form graph and compute diffusion maps

1

2

3

Gaussian kernel in dimensionality reduction (Short history)

The assumption that high dimensional data reside on or near a low dimensional manifold inspired many theoretical and experimental results.

- Schölkopf and Samola used uses the gaussian kernel with no normalization for non-linear PCA .
- Belkin and Niyogi normalize the gaussian kernel to be the laplacian of a graph defined on the data.
- Coifman and Lafon further normalize for non-uniform sampling from the manifold.

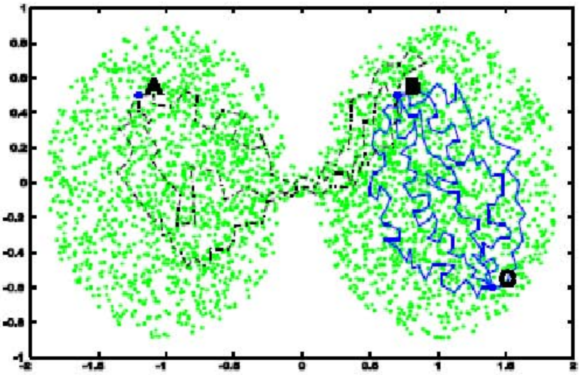
Given a set of n input vectors $x_i \in \mathbb{R}^d$

1. $K_0(i,j) \leftarrow e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}}$
2. $p(i) \leftarrow \sum_{j=1}^{n} K_0(i,j)$ approximates the density at x_i
3. $\widetilde{K}(i,j) \leftarrow \frac{K_0(i,j)}{p(i)p(j)}$
4. $d(i) \leftarrow \sum_{j=1}^{n} \widetilde{K}(i,j)$
5. $K(i,j) \leftarrow \frac{\widetilde{K}(i,j)}{\sqrt{d(i)}\sqrt{d(j)}}$
6. $USU^T = K$ (by SVD of K)

Stages 2 and 3 normalize for density; stages 4 and 5 perform the graph laplacian normalization. In limit $n \to \infty$, and $\sigma \to 0$

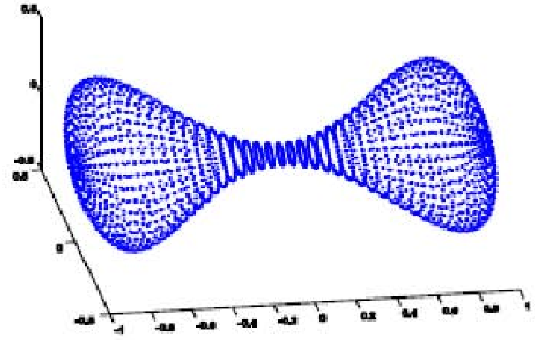
- K converges to a conjugate to the diffusion operator Δ.
- The functions $\varphi_k(x) = u_k(x)/u_0(x)$ converge to the eigenfunctions of Δ on M.

The quantity $D_m(x, y)$ is a distance between points that measures the connectivity of x and y in the data. More robust than geodesic distance

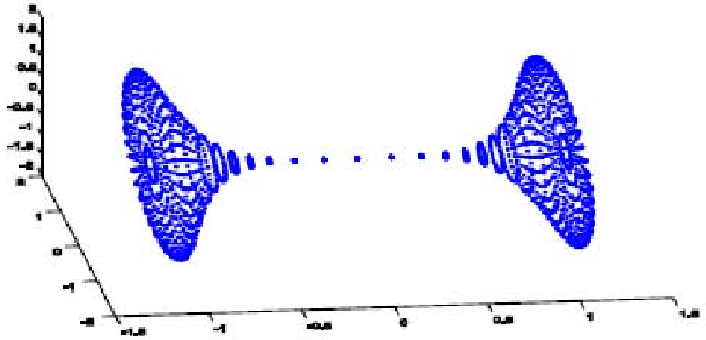


The maps $\Phi^{(m)}$ give a new representation of the data as points in a Euclidean space.

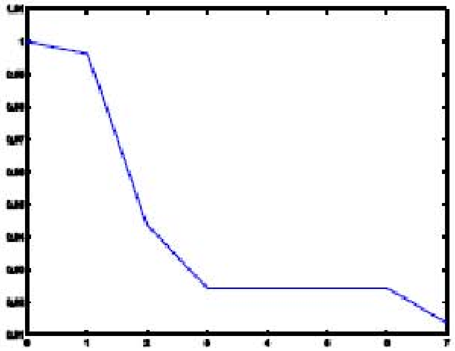
©Steven W. Zucker



-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

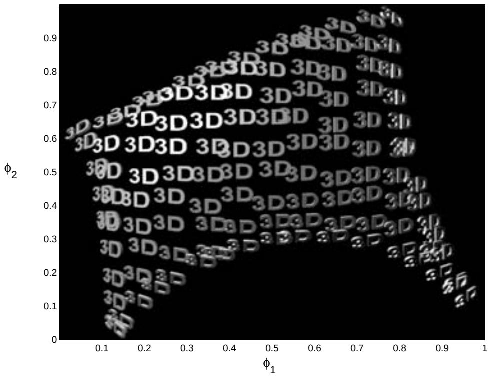


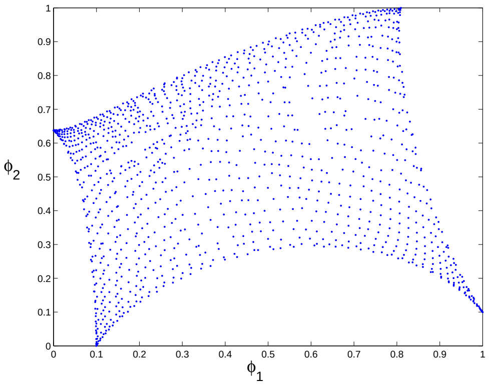
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5



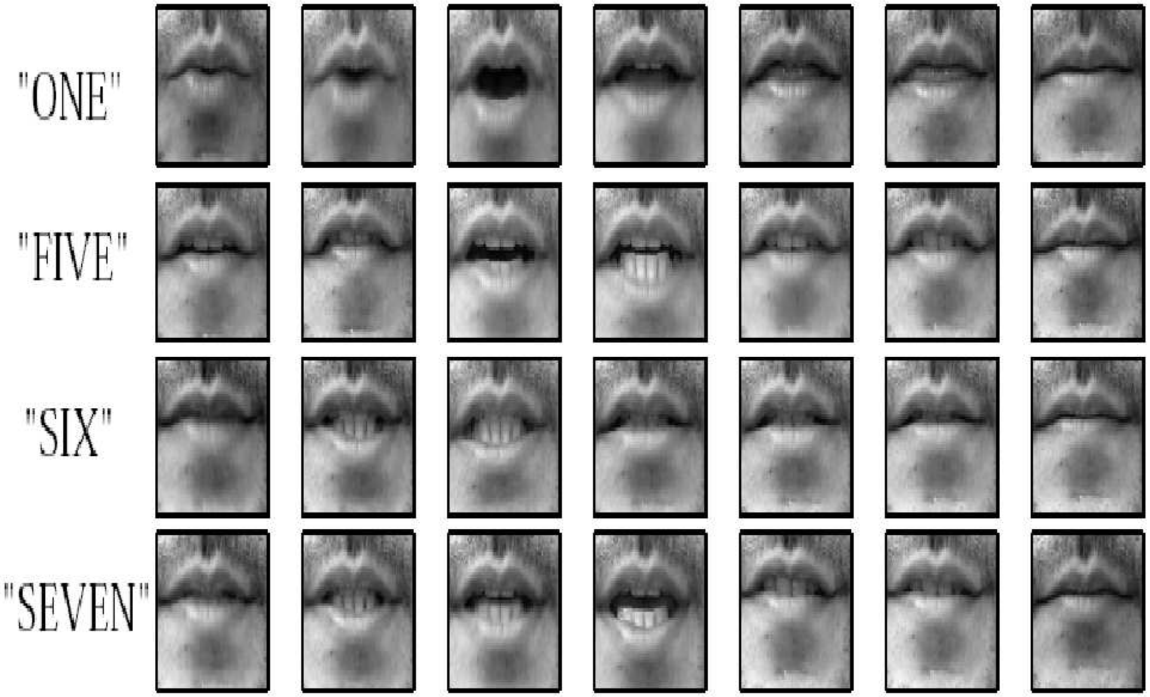
1
0
1
2
3
4
5
6
7

Diffusion Maps Reveal “Manifold”

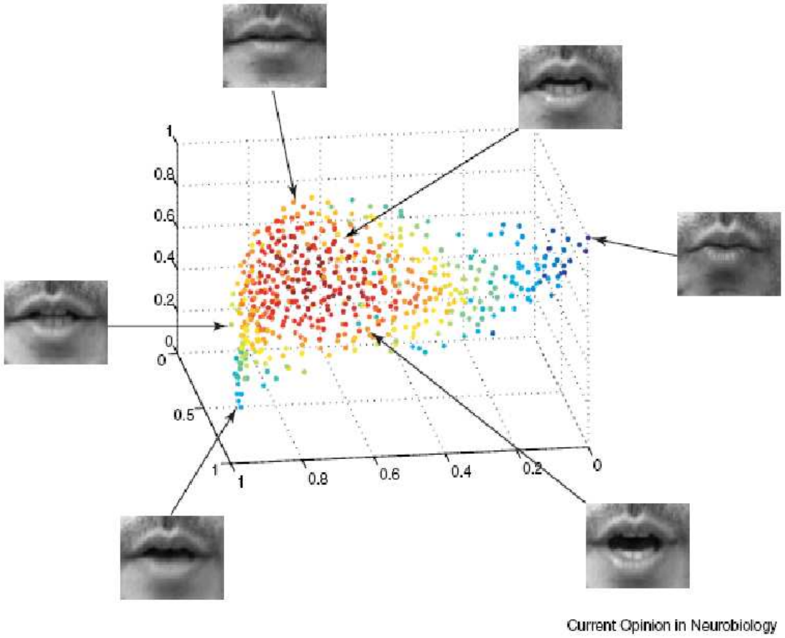




Reading Lips



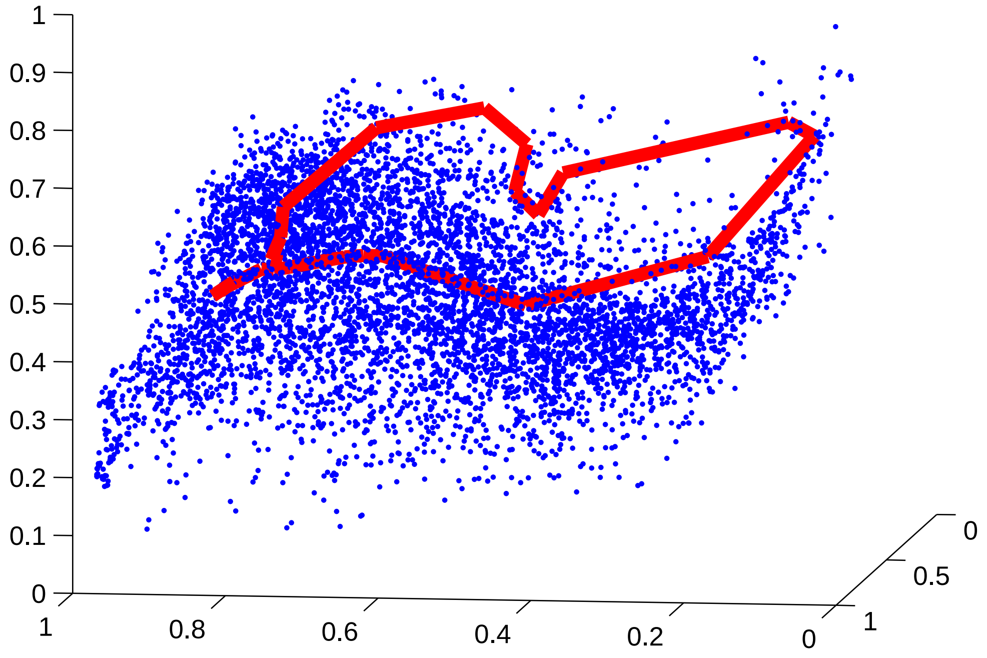
Diffusion Maps Reveal “Manifold”



Current Opinion in Neurobiology

©Steven W. Zucker

Digit Trajectories over “Manifold”



Visual Data Classifier

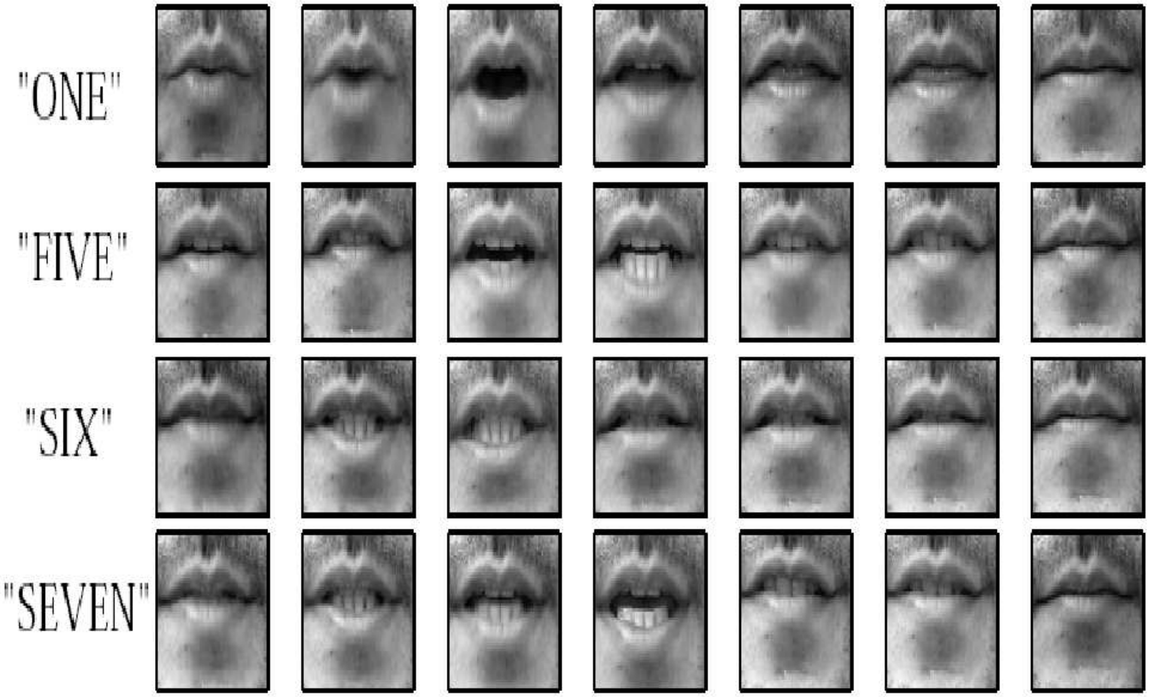
- View digit words as a trajectory in the diffusion space
- Word recognition is identifying trajectories
- Classifier: compare new trajectory to collection of labeled (training) trajectories.
- Use symmetric Hausdorff distance between two sets Γ_1 and Γ_2, defined as

$$d(\Gamma_1, \Gamma_2) = \max\left\{ \max_{x_2 \in \Gamma_2} \min_{x_1 \in \Gamma_1} \{\|x_1 - x_2\|\}, \max_{x_1 \in \Gamma_1} \min_{x_2 \in \Gamma_2} \{\|x_1 - x_2\|\} \right\}. \tag{1}$$

Visual Data Classification

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.90**	0	0	0.01	0	0	0.08	0	0	0
one	0	**0.99**	0	0	0	0	0	0	0.01	0
two	0.04	0.01	**0.90**	0.03	0.02	0	0	0	0	0
three	0	0	0.01	**0.94**	0	0	0.01	0.02	0.01	0
four	0.01	0	0	0.05	**0.93**	0	0	0	0	0
five	0	0	0	0	0	**0.81**	0.01	0.16	0	0.01
six	0.07	0	0	0.01	0	0	**0.87**	0.03	0.01	0.01
seven	0.03	0	0	0.04	0	0.07	0.05	**0.74**	0.04	0.02
eight	0	0	0	0	0.02	0.03	0	0.03	**0.75**	0.16
nine	0	0	0	0	0	0	0	0.04	0.14	**0.82**

Note Similarities between *Six* and *Seven*



Audio Data Classification ($n = 10$)

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.75**	0	0.04	0	0.01	0.01	0.06	0.08	0.05	0
one	0	**0.94**	0	0	0	0.03	0	0	0	0.02
two	0.02	0	**0.87**	0.04	0.01	0	0.01	0	0.03	0.02
three	0.01	0	0.03	**0.90**	0.02	0.01	0	0	0.01	0.01
four	0.01	0	0	0.02	**0.96**	0	0	0	0	0.01
five	0.01	0.01	0	0.06	0	**0.86**	0	0.01	0.01	0.03
six	0	0	0	0	0.01	0	**0.93**	0.05	0	0
seven	0.05	0	0	0	0	0	0.14	**0.81**	0.01	0
eight	0.02	0	0.04	0.02	0	0.02	0	0.07	**0.80**	0.03
nine	0	0.01	0	0.01	0.01	0.04	0	0	0.01	**0.92**

Multisensor Embedding For Sensor Fusion

- Starting with K input sources $\Omega_k = \{y_1^k, ..., y_n^k\}$, $k = 1...K$.
- Compute the Laplace-Beltrami embeddings of $\{\Omega_k\}$, denoted $\Phi_k^{m_k}$, where m_k is the dimensionality of the embedding of the k'th channel.
- Compute the unified coordinates set $\widehat{\Omega} = \{z_1, ..., z_n\}$ by appending the embeddings of each input sensor

$$z_i = \{\phi_i^1, \ldots, \phi_i^K\}, i = 1...n, \ , k = 1...K.$$

Audio and Visual Data Classification ($n = 5 + 5$)

	“0”	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
zero	**0.90**	0.00	0.00	0.00	0.00	0.00	0.06	0.04	0.00	0.00
one	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
two	0.00	0.00	**0.96**	0.01	0.02	0.00	0.00	0.00	0.00	0.00
three	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.00	0.00	0.01
four	0.00	0.00	0.00	0.04	**0.96**	0.00	0.00	0.00	0.00	0.00
five	0.00	0.00	0.00	0.00	0.00	**0.97**	0.00	0.00	0.02	0.01
six	0.06	0.00	0.00	0.00	0.00	0.00	**0.90**	0.04	0.00	0.00
seven	0.03	0.00	0.00	0.00	0.00	0.00	0.03	**0.93**	0.00	0.00
eight	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.01	**0.95**	0.03
nine	0.00	0.01	0.00	0.00	0.00	0.01	0.00	0.00	0.02	**0.96**

Summary Classification

Channel type	**“0”**	**“1”**	**“2”**	**“3”**	**“4”**	**“5”**	**“6”**	**“7”**	**“8”**	**“9”**
Audio	0.75	0.94	0.87	0.90	**0.96**	0.86	**0.93**	0.81	0.80	0.92
Visual	**0.90**	**0.99**	0.90	0.94	0.93	0.81	0.87	0.74	0.75	0.82
Combined	**0.90**	**0.99**	**0.96**	**0.99**	**0.96**	**0.97**	0.90	**0.93**	**0.95**	**0.96**

"The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL/SNAT (now AFRL/RYAT) or the U.S. Government." "This material is based on research sponsored by AFRL/SNAT under agreement number FA8650-05-1-1800 (BAA 04-03-SNK Amendment 3). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon."

Final Presentation - Part 2[a]
Diffusion Maps and Geometric Harmonics for ATR

FA8650-05-1-1800

23 Nov 2004 - 31 Oct 2007

Steven W. Zucker & Ronald R. Coifman

Department of Computer Science and Program in Applied Mathematics

Yale University

[a] The presentation is ©2007 Steven W. Zucker

Psychological Questionnaires

Answer by YES or NO

Group A

- I find it hard to wake up in the morning.
- I'm usually burdened by my tasks for the day.
- I love dancing.

What about **Group B**?

- I like poetry.
- I might enjoy being a dog trainer.
- I read the newspaper every day.

Group A are questions like the ones in the MMPI-2 test, aimed at estimating depression

- I find it hard to wake up in the morning. (yes)
- I'm usually burdened by my tasks for the day. (yes)
- I love dancing. (no)

In the MMPI-2 a (raw) score is the sum of ”correct answers”.

Group B, designed to test for other conditions, seem unrelated to depression.

- I like poetry. (?)
- I might enjoy being a dog trainer. (?)
- I read the newspaper every day. (?)

Questions:

- Are **Group B** answers informative about depression?
- If so, can incomplete questionnaires be scored correctly?
- Is the space of answers structured? and How?

Answering the latter suggests an approach to the former.

- MMPI-2 structure
- Manifold learning

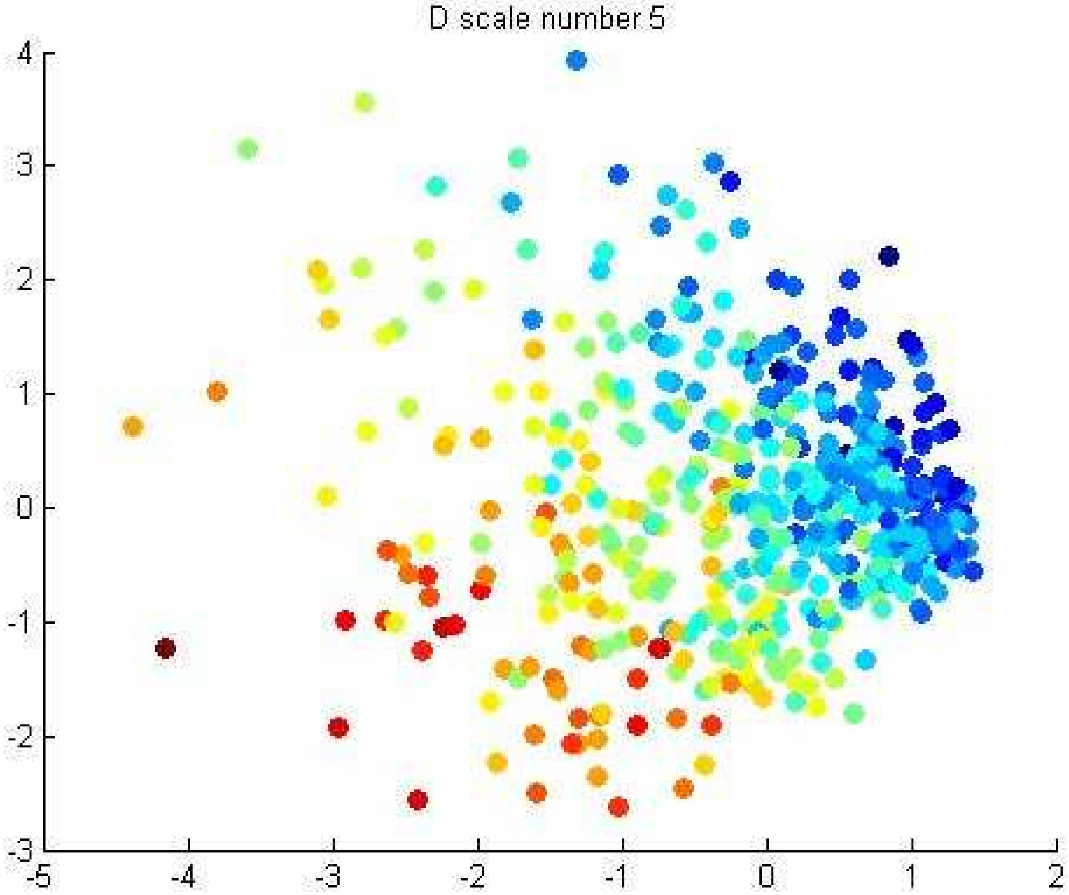
MMPI-2 and the diffusion framework

- Ambient space: $x \in 567$ dimensions (yes/no answers → ±1).
- A set of responses x_i lie on or near a low dimensional manifold M in $\mathbb{R}^d$
- M is sufficiently sampled with some density p by the training set. For a given function g and a compact subset of $\mathbb{R}^d$,Ω:

$$\sum_{x_i \in \Omega} g(x_i) \approx \int_{\Omega \cap M} g(x)p(x)d\Omega \qquad (1)$$

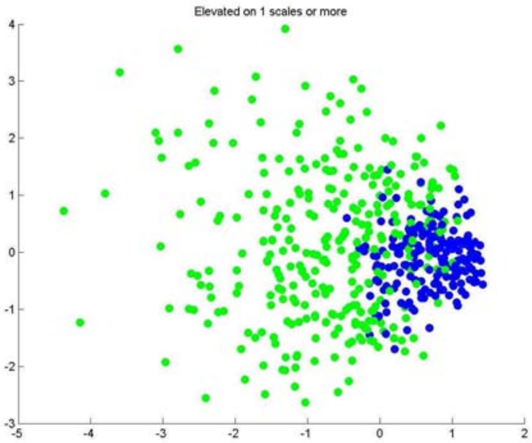
- scales: functions on the answer vectors $f_{diagnosis}(x) : \mathbb{R}^d \rightarrow \mathbb{R}$. summation of "correct answers".
- diagnosis $\in$ { anxiety, depression, $\ldots$, hysteria }.
- The scoring function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is smooth on M.

Scales as Functions on Data Points: Depression



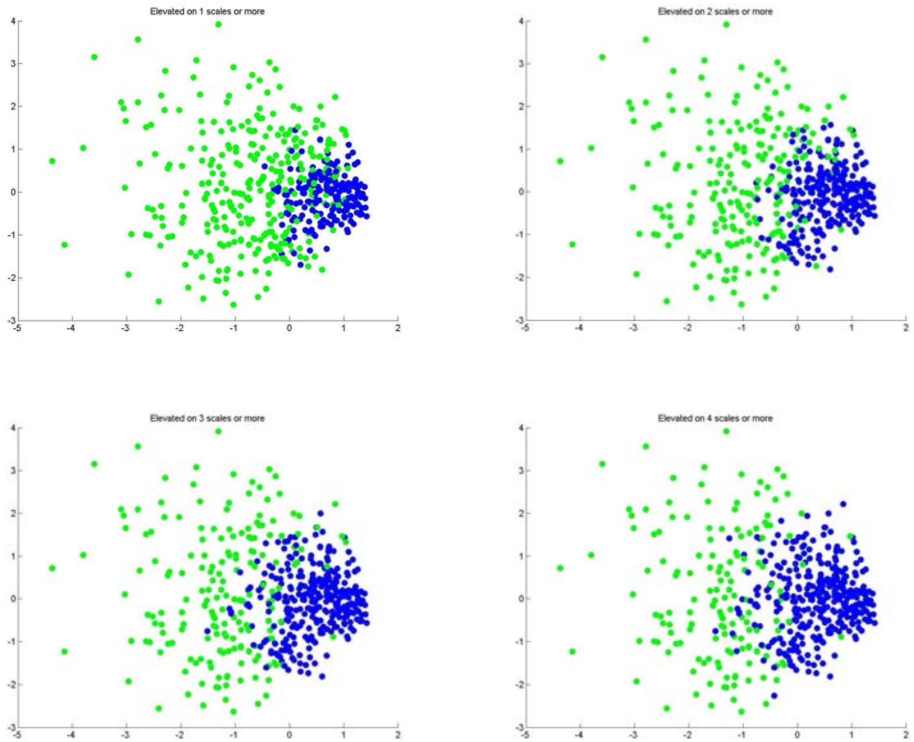
©Steven W. Zucker

Elevated on One Scale

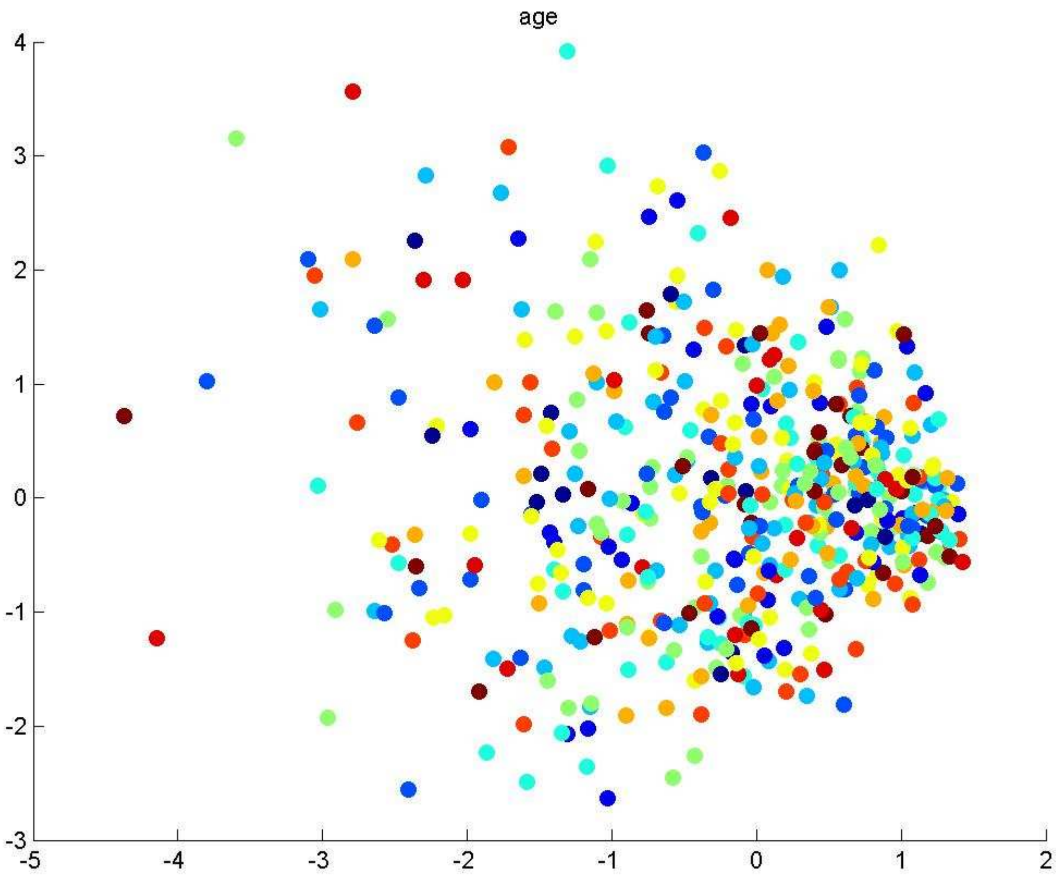


- Green: pathological
- Blue: Normal

Elevated on Multiple Scales



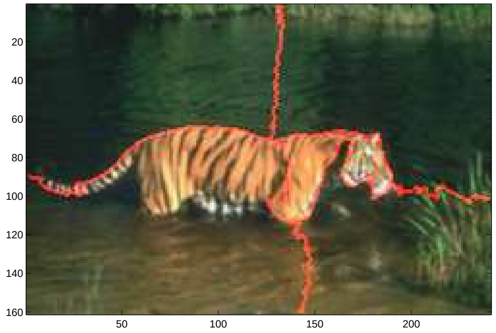
Age Scale Not Informative



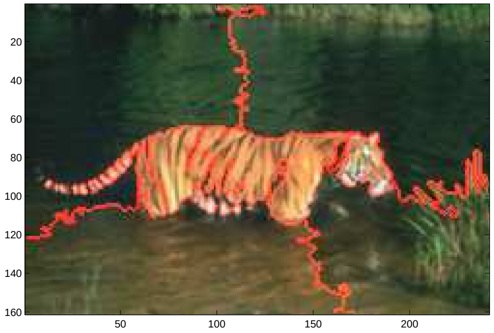
©Steven W. Zucker

- When a function maps regularly onto data points, it can be extended to new data points.
- Fill-in missing data.
- check consistency of data.
- find ”outliers”

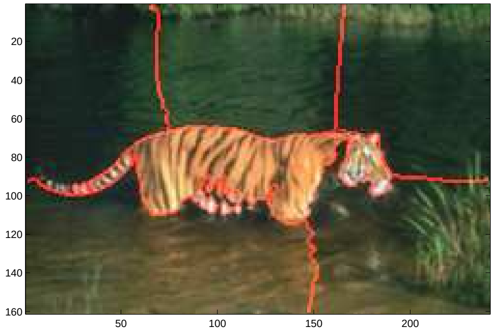
Sensor Integration for Segmentation-1



Shi/Malik/intensity

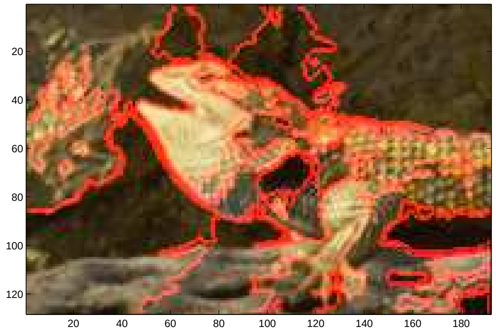


RGB

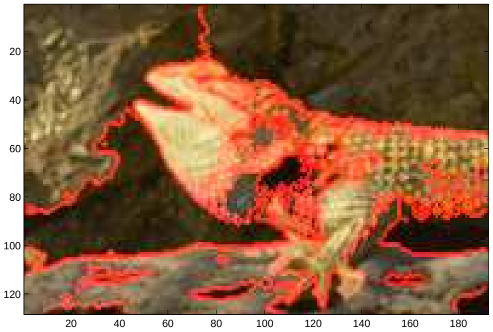


Combined

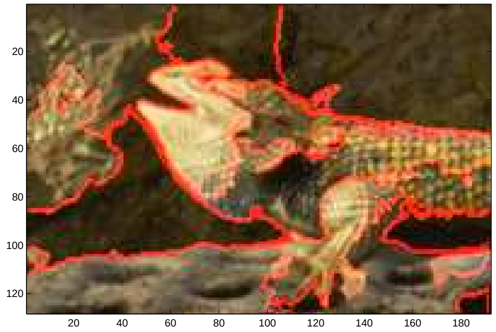
Sensor Integration for Segmentation-2



texture



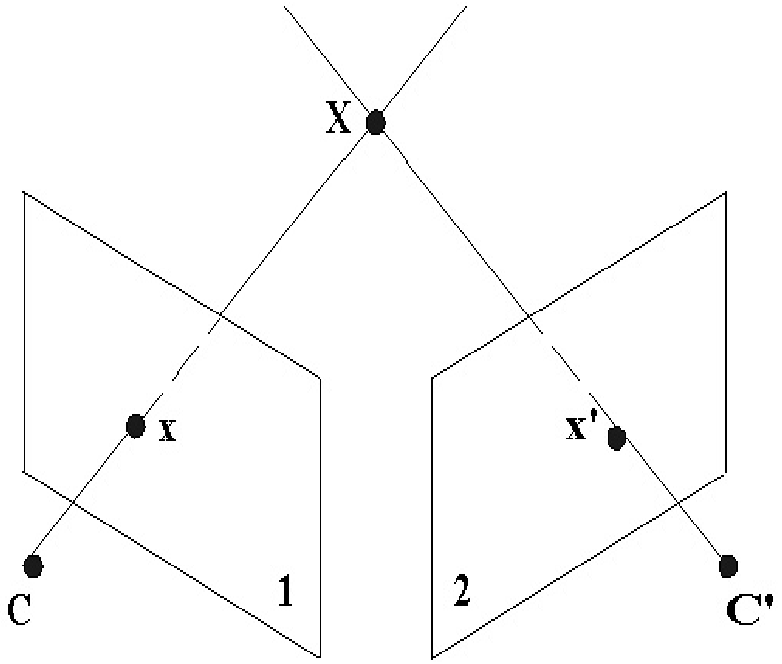
RGB



Combined

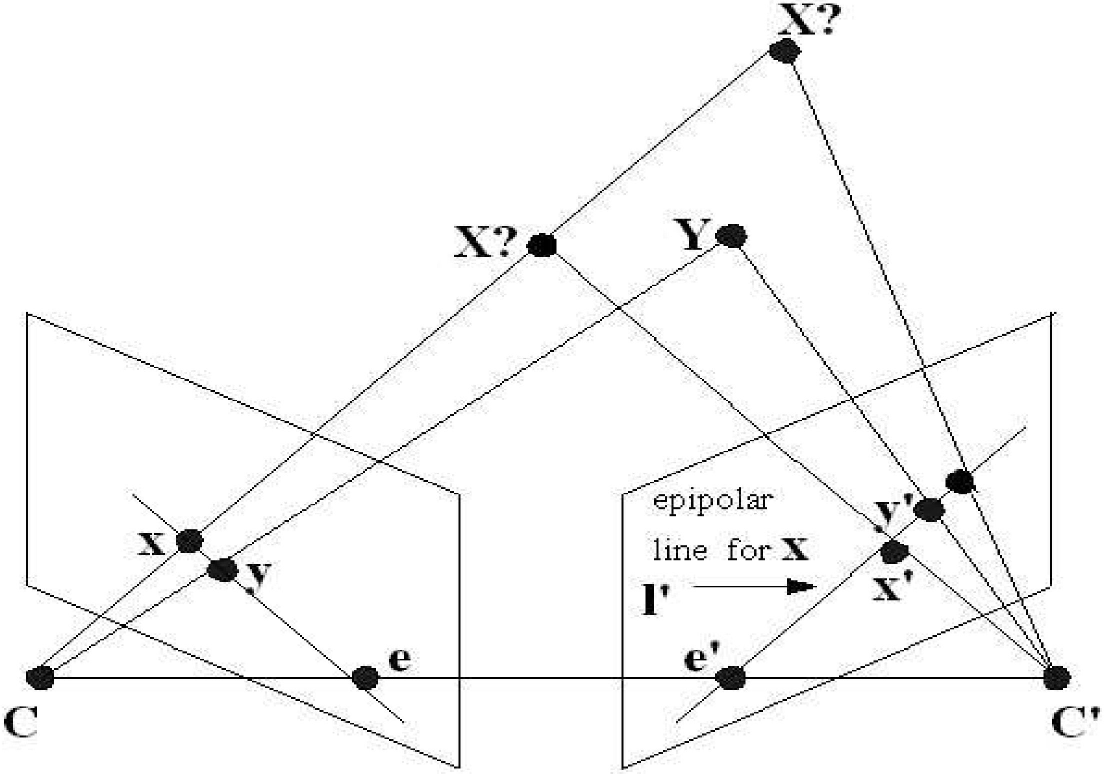
Stereo Correspondence Problem

Recover the depth information from the disparity (difference of image coordinates) of the corresponding image points.



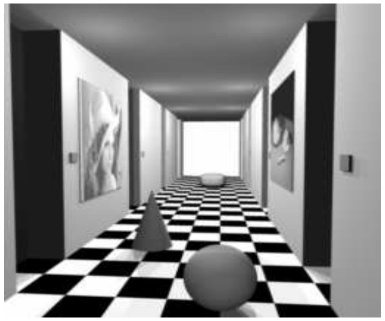
Matching Constraints in Stereo Correspondence

- Epipolar Constraint (geometric constraint).
- Ordering Constraint (heuristic constraint).



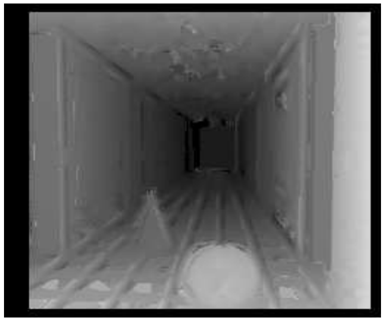
Frontal Parallel Plane Assumption in Stereo Correspondence

- Assume surface is parallel (i.e. at constant depth) to the image pairs.
- slide window; select position s.t. max SSD

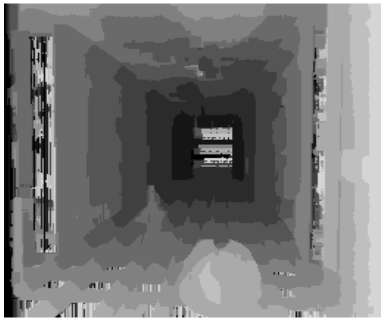


(left)

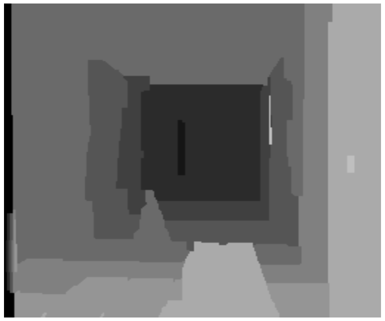
(right)



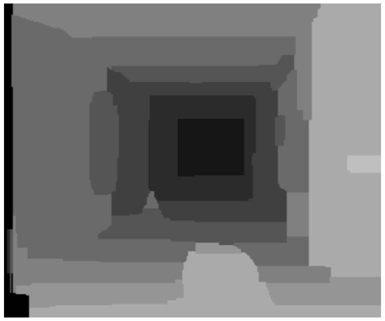
(our result)



(SSD)

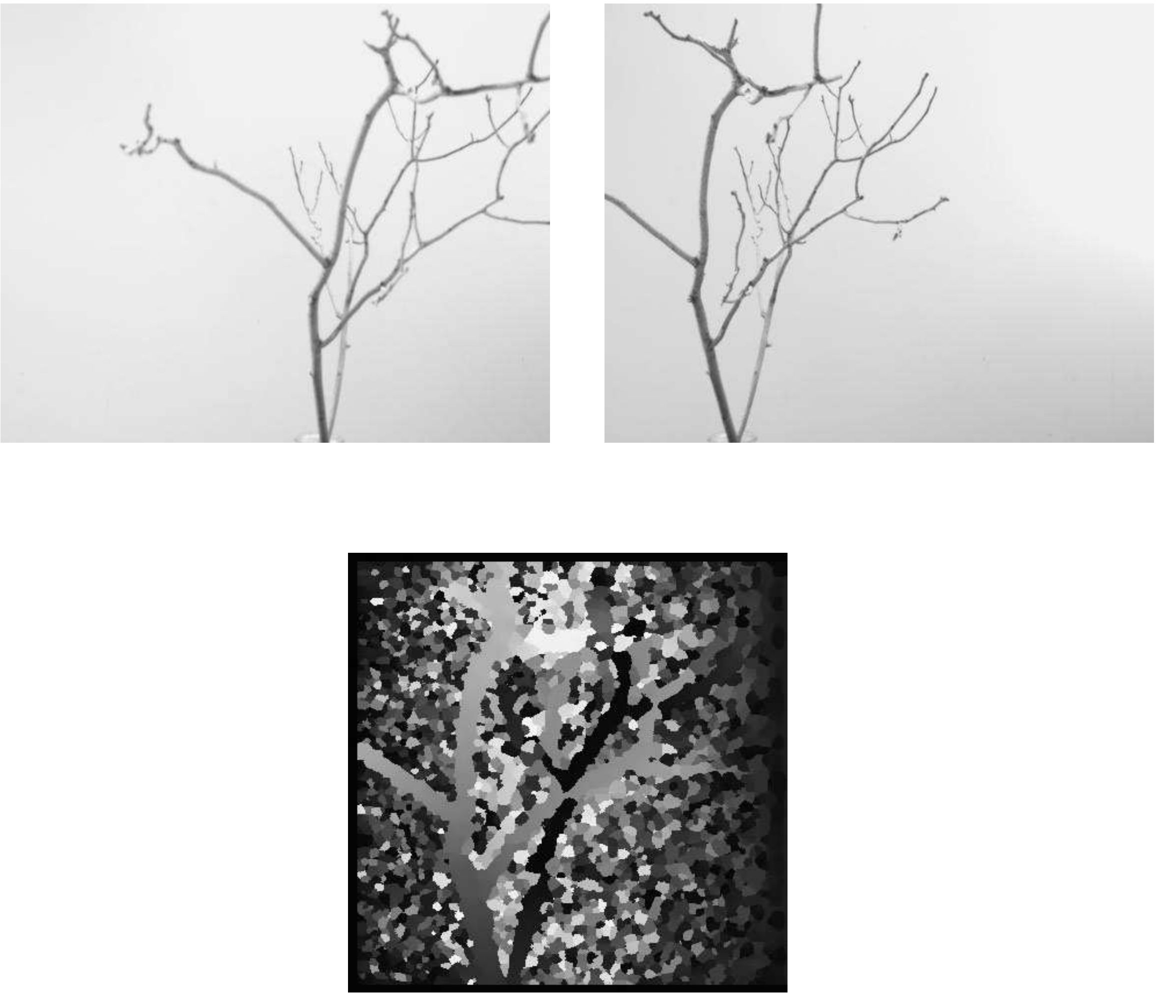


(graph cut)



(belief prop)

Display (brightness = depth)



Zitnick and Kanade, PAMI 2000.

Next Step: Use contextual information *geometrically* ("directed diffusion") in stereo correspondence

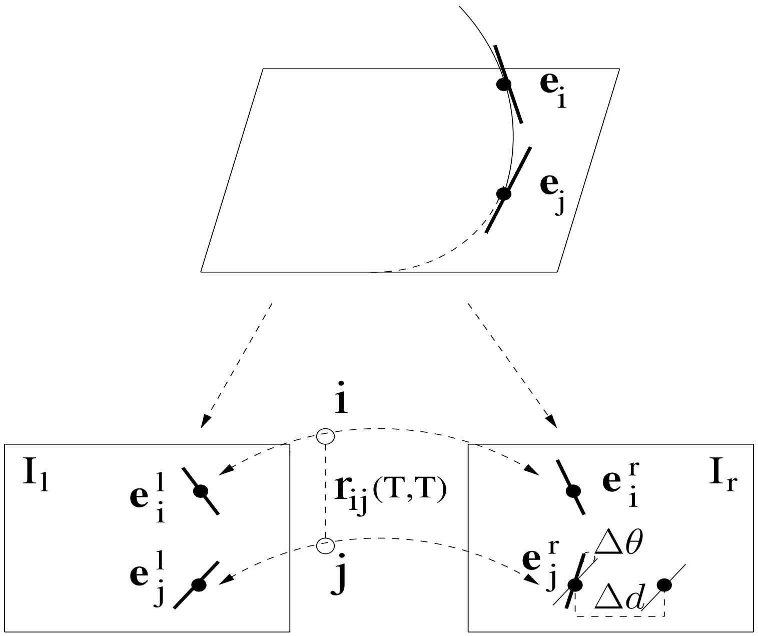
Road map:

- Space Curves — Frenet
- Smooth Surfaces — Cartan

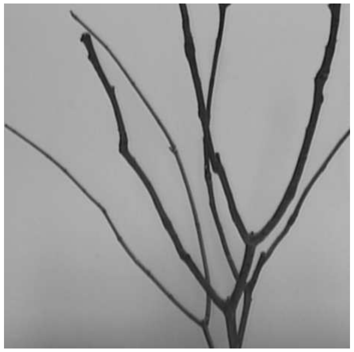
Imposing Geometric Constraints Over Neighboring Matching Pairs

- Build a local model (Frenet approximation) for every (possible) curve point j.
- Predict the Position and the Frenet frame at a nearby position i. Compare with the measurements at i. They should agree if i comes from the same curve as j.
- Enforce such consistency and only retain those that are compatible with their neighbors.

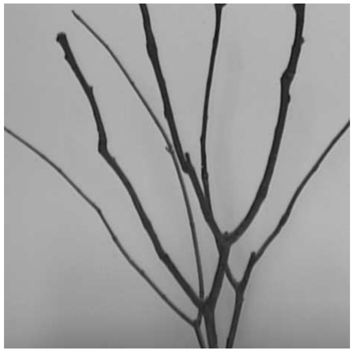
- Each space tangent projects to a pair of image tangents.
- Both positional disparity (Δd) and orientation disparity ($\Delta\theta$) used.



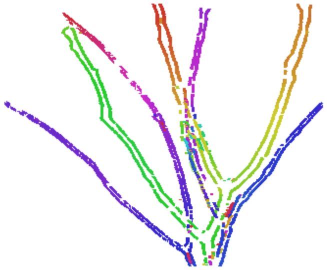
Results



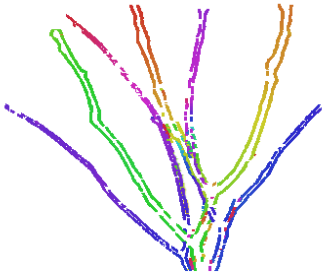
(left)



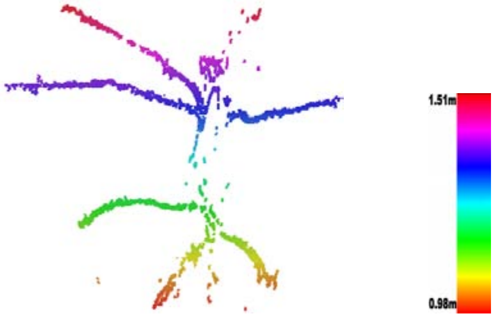
(right)



(view 1)

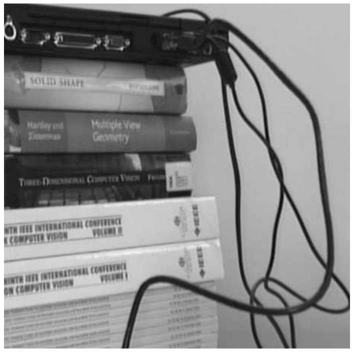


(view 2)

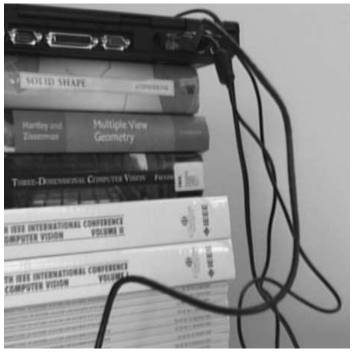


(scale: m)

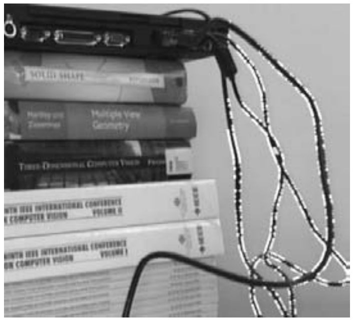
Results



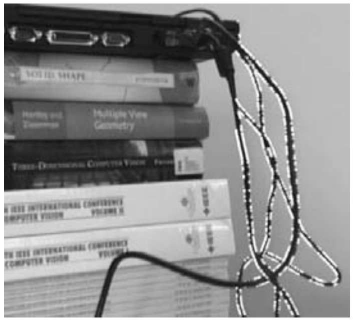
(left)



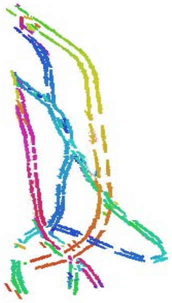
(right)



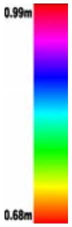
(matched-L)



(matched-R)



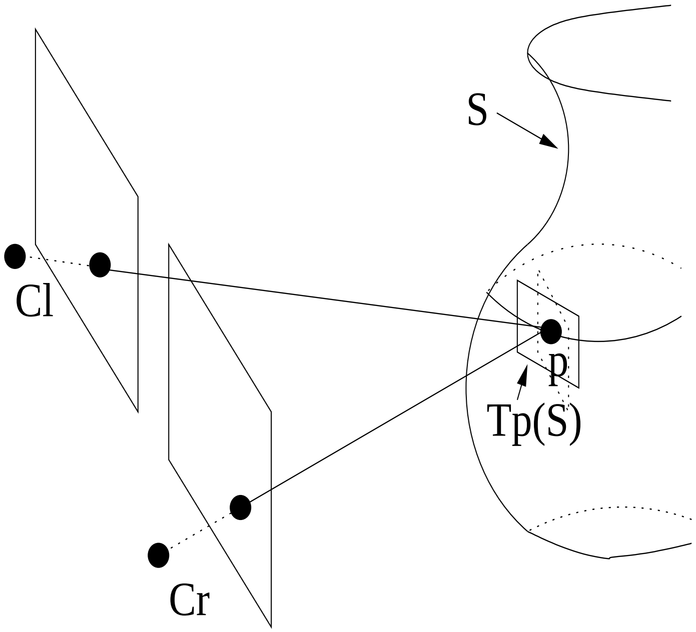
(result)



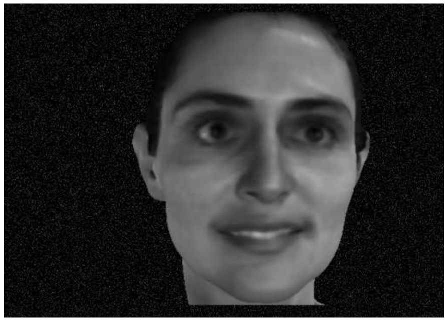
(scale: m)

Stereo Correspondence for Surface Reconstruction

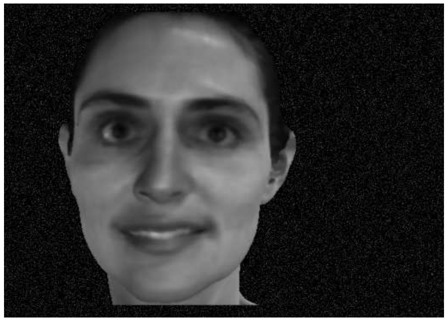
- Goal: Dense reconstruction of smooth surfaces.
- Observe: Tangent plane $T_p(M)$ (in solid lines) deviates from the frontal parallel plane (in dotted lines).



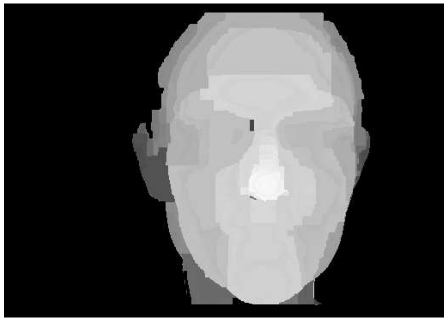
Results



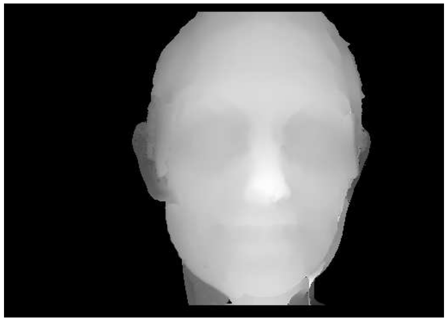
Left (reference) image



Right image

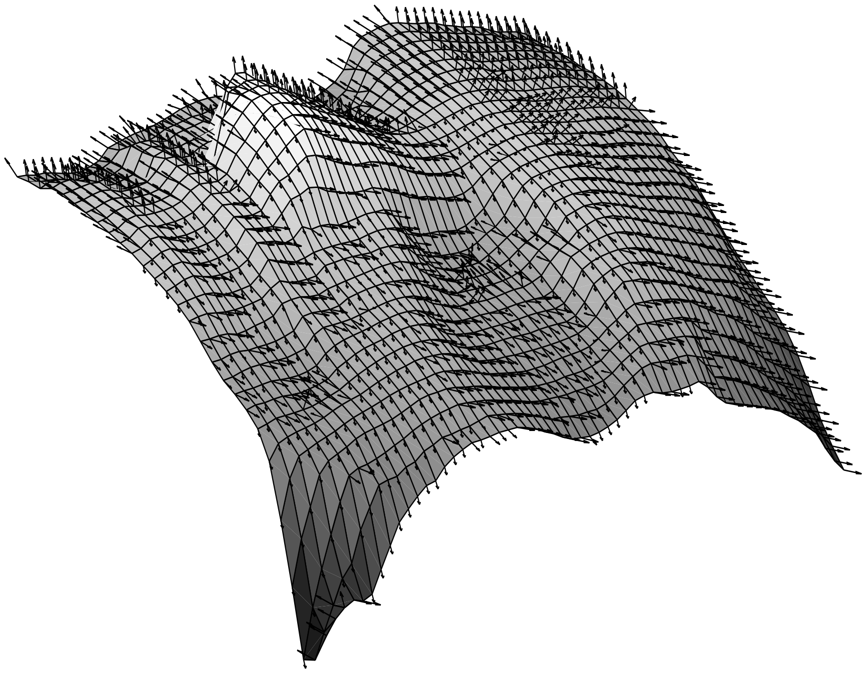


GC+Subpixel



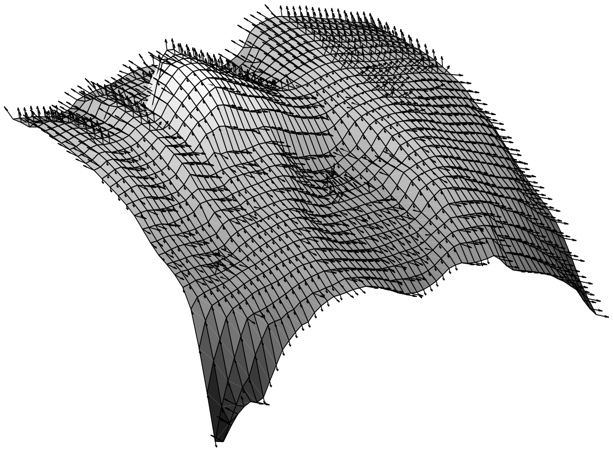
Our Result

Tammy-Normals

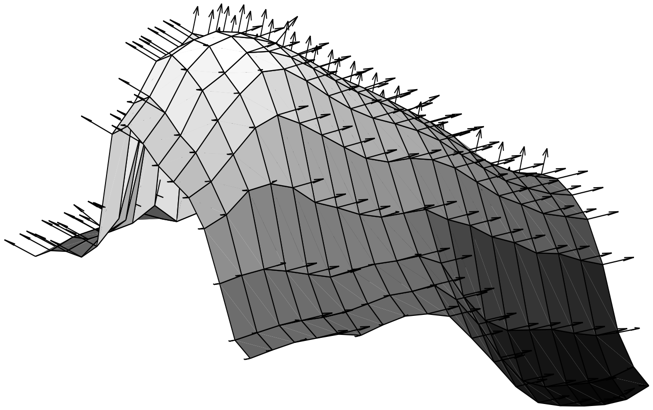


©Steven W. Zucker

Tammy-Zoom



Surface Normals



Zoom

Organization of Spectral Information

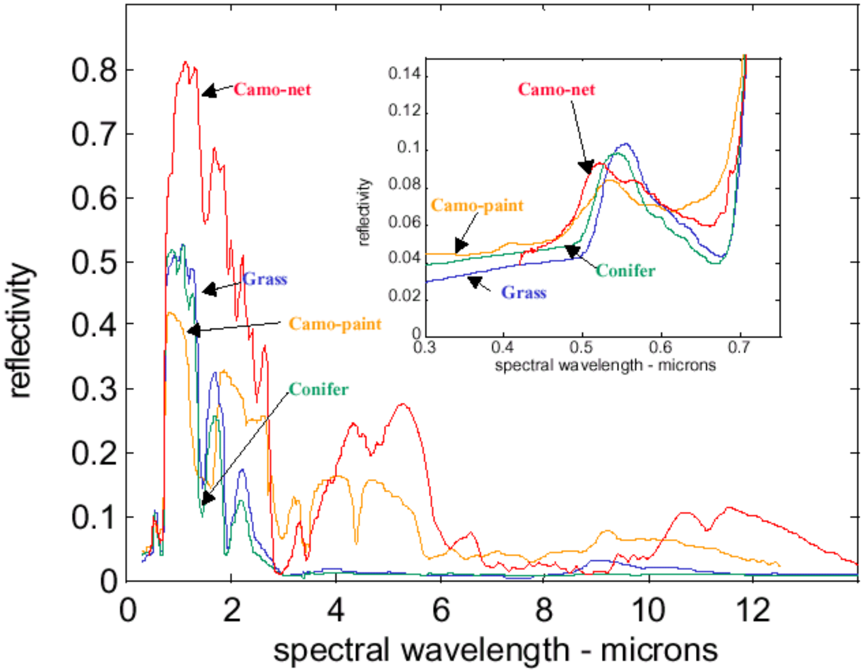
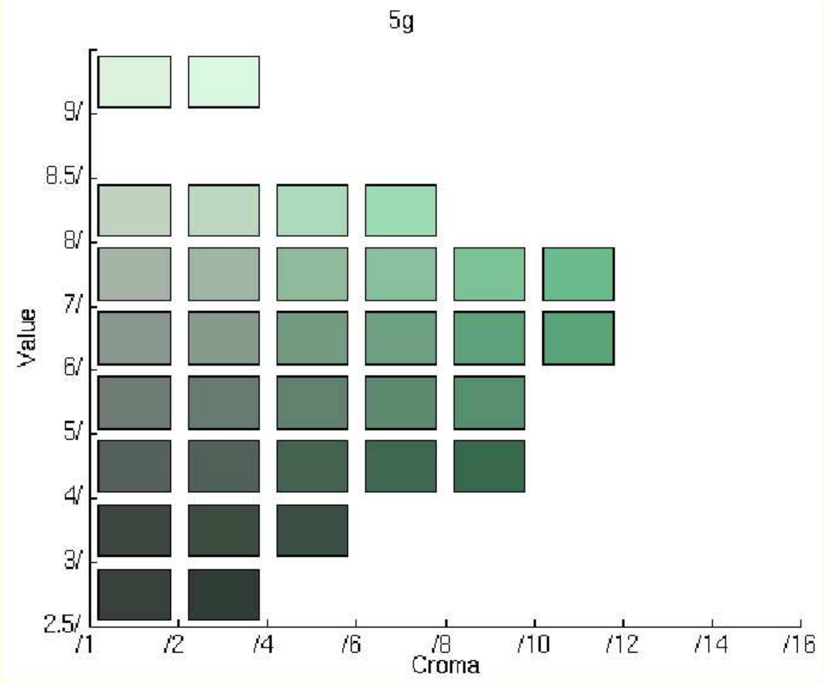


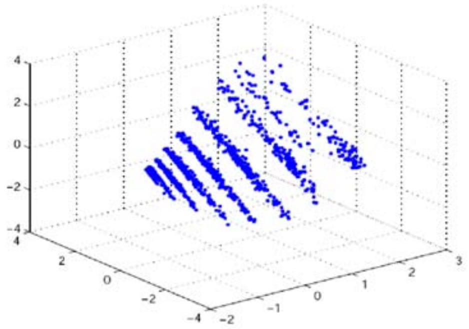
Fig. 1 Reflectivity of camouflage and of conifer trees and grass, showing that the camouflage is relatively ineffective in the 4-6 μm and 10-12 μm regions (after D. Scribner (NRL)

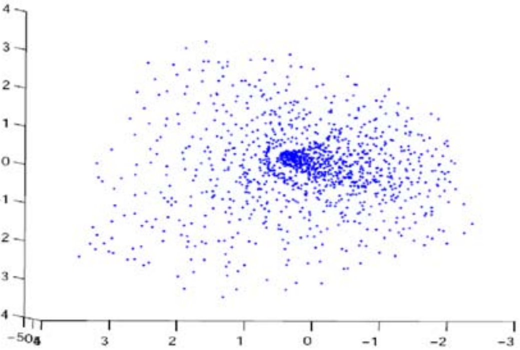
Munsell Color Patches

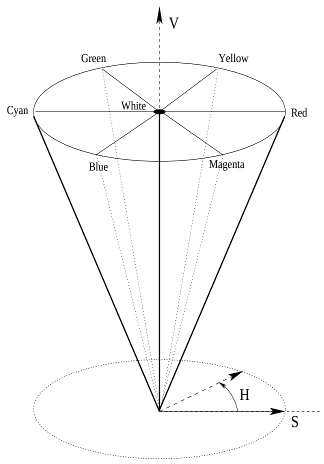


The Matlab image of the page 5G of the Munsell Book of Color. RGB colors are calculated from the AOTF measured spectra of the Munsell colors.

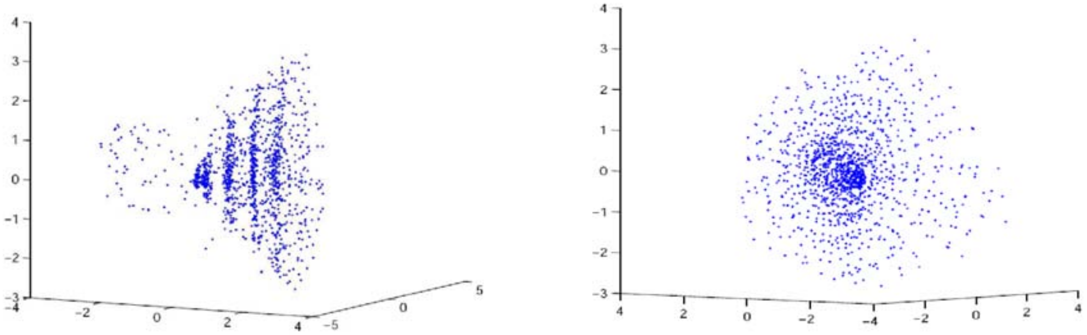
Color Diffusion Map



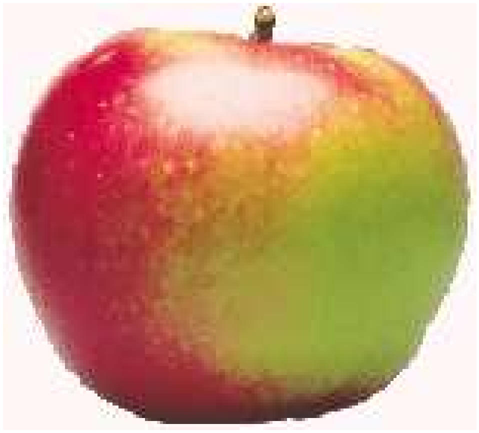


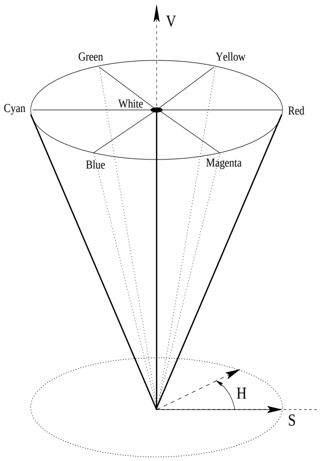


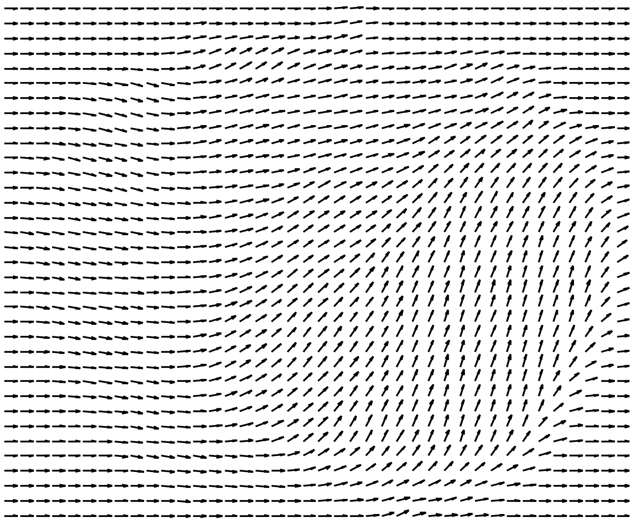
Color Diffusion Map through Retinal Pigments



Color and geometry







Future Work:

- Abstract Features (in *feature space*
- Geometry of fusion of boundary/texture/color data for improved segmentation and classification.
- Fusion of spectral and spatial data
- (Fusion of left/right stereo pairs): feedback to image segmentation following biological model.